AIRCRAFT COMMAND INTERFACE FOR A REAL-TIME TRAJECTORY PLANNING SYSTEM

by

JAMES RICHARD ALEXANDER

Bachelor of Science in
Aerospace Engineering
The University of Michigan, Ann Arbor (1986)

SUBMITTED TO THE DEPARTMENT OF
AERONAUTICS AND ASTRONAUTICS IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

November 1987

Signature of Author _____
Department of Aeronautics and Astronautics
November 13, 1987

Certified by _____
Professor Wallace E. Vander Velde
Thesis Supervisor

Accepted by _____
Professor Harold Y. Wachman
Chairman, Deparmental Graduate Committee

AIRCRAFT COMMAND INTERFACE FOR A REAL-TIME
TRAJECTORY PLANNING SYSTEM

by
JAMES RICHARD ALEXANDER

Submitted to the Department of Aeronautics and
Astronautics on November 13, 1987, in partial fulfillment
of the requirements for the degree of
Master of Science in Aeronautics and Astronautics

ABSTRACT

Due to the increasing complexity of vehicle missions, computer-based systems capable of providing autonomous control for unmanned vehicles or semi-autonomous control for manned vehicles are needed. Trajectory planning algorithms in these systems compute trajectories for a vehicle based upon desired mission objectives, threats, terrain constraints, and other factors. The development of trajectory planning algorithms is being actively addressed in the research community. Trajectories created by these systems are not in a form that can be interpreted directly by the vehicle control system. Therefore, a command interface must be developed that can interpret the trajectories and correspondingly generate commands for the vehicle control system based upon the current state of the vehicle.

The objective of this research is to investigate the nature of such a command interface, define the necessary functions the interface must execute, and test interface models using simulations. Approaches to a specific interface for an aircraft flight control system have been defined and developed. A simulation has been prepared to test the overall system feasibility and performance and to investigate implementation options.

Thesis Supervisor: Wallace E. Vander Velde
Title: Professor of Aeronautics and Astronautics

AIRCRAFT COMMAND INTERFACE FOR A REAL-TIME TRAJECTORY PLANNING SYSTEM

by

JAMES RICHARD ALEXANDER

NOVEMBER 1987

Acknowledgements

I would like to extend sincere thanks to my research advisor, Professor Wallace E. Vander Velde. He has given me the opportunity to experience MIT, and his incredible insight and experience have made this research possible.

A large amount of credit goes to my beautiful wife, Tia Marie. I can't begin to tell you how happy I am that she stuck around until this thing was finished. Thanks so much for your support and enthusiasm.

I would like to thank all the fellow grad students who have made my experience at MIT all worthwhile. To the gang at Ashdown, I'll never forget all the great times; and to the Flunkies . . . well, I couldn't forget you guys even if I tried.

Finally, my parents deserve more thanks than I could ever give them. Their support through the years has kept me going and has helped me achieve more than I have ever dreamed. Thanks, mom and dad.

This research was sponsored by The Charles Stark Draper Laboratory, Inc. under IR&D numbers 18211 and 18922. Many thanks to Paul Motyka and the System Sciences Division for their input to this work.

Table of Contents

Abstract 2
Acknowledgements 3
Table of Contents 4
List of Figures 6
List of Tables 8
Glossary of Symbols 9

1. Introduction 11

1.1 Nature of the Problem 11
1.2 Research Objectives 12
1.3 Thesis Organization 13

2. Trajectory Planner and Aircraft Flight Control System Models 14

2.1 The Trajectory Planner 14
2.3 Aircraft Flight Control System Model 16

3. The Aircraft Command Interface 18

3.1 Introduction 18
3.2 Creating Smooth Desired Trajectories 19
3.2.1 Unconstrained Least-Squares Curve Fitting 21
3.2.2 Constrained Least-Squares Curve Fitting 22
3.3 Command Generation Algorithm 23
3.3.1 Acceleration Commands Based on Tracking Error 24
3.3.2 Acceleration Commands Based on Nominal Acceleration 25
3.3.3 Normal Acceleration and Bank Angle Commands 28
3.4 Simulating the System Dynamics 29

4. Simulation Results 33

4.1 Parameter Choices in the Simulation 33
4.2 Unconstrained Curve Fitting Results 35

Table of Contents (continued)

4.3 Constrained Curve Fitting Results 37
4.4 Command Limiting 40
4.5 Choice of Sign For Normal Acceleration Commands 45

5. Conclusions and Recommendations 73

5.1 Conclusions 73
5.2 Recommendations 75

References 77
Appendix A 78

List of Figures

1.1: Schematic of a Computer-Based Vehicle Controller 11

2.1: Model for Each Channel of the Aircraft FCS 17

3.1: Definition of Notation 30

3.2: Subsets of Waypoints for Unconstrained Least-Squares Curve Fits . 30

3.3: Model Used to Determine K_1 and K_2 31

3.4: Unit Vector Orientation for General Curvilinear Motion 31

3.5: Generation of Normal Acceleration and Bank Angle Commands 32

3.6: Simulation Model of Aircraft Command Interface 32

4.1: Use of Unconstrained Curve Fits to Represent the Desired Trajectory 35

4.2: X-Y Reference Frame Coordinates for the Half-Circle Waypoint Set 48

4.3: Aircraft Altitude vs Time for the Half-Circle Waypoint Set Using Unconstrained First-Order Curve Fits in Subsets of Six . . . 49

4.4: Aircraft Tracking Error vs Time for the Half-Circle Waypoint Set Using Unconstrained First-Order Curve Fits in Subsets of Six . . 50

4.5: Aircraft Normal Acceleration vs Time for the Half-Circle Waypoint Set Using Unconstrained First-Order Curve Fits in Subsets of Six 51

4.6: Aircraft Bank Angle vs Time for the Half-Circle Waypoint Set Using Unconstrained First-Order Curve Fits in Subsets of Six 52

4.7: Aircraft Altitude vs Time for the Half-Circle Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index . . . 53

4.8: Aircraft Tracking Error vs Time for the Half-Circle Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index . . 54

4.9: Aircraft Normal Acceleration vs Time for the Half-Circle Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index 55

4.10: Aircraft Bank Angle vs Time for the Half-Circle Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index 56

4.11: Aircraft Altitude vs Time for the Half-Circle Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index . . . 57

List of Figures (continued)

4.12: Aircraft Tracking Error vs Time for the Half-Circle Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index . . 58

4.13: Aircraft Normal Acceleration vs Time for the Half-Circle Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index 59

4.14: Aircraft Bank Angle vs Time for the Half-Circle Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index 60

4.15: X-Y Reference Frame Coordinates for the Sine Curve Waypoint Set 61

4.16: Aircraft Altitude vs Time for the Sine Curve Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index 62

4.17: Aircraft Tracking Error vs Time for the Sine Curve Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index . . . 63

4.18: Aircraft Normal Acceleration vs Time for the Sine Curve Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index 64

4.19: Aircraft Bank Angle vs Time for the Sine Curve Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index 65

4.20: Aircraft Altitude vs Time for the Sine Curve Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index 66

4.21: Aircraft Tracking Error vs Time for the Sine Curve Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index . . . 67

4.22: Aircraft Normal Acceleration vs Time for the Sine Curve Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index 68

4.23: Aircraft Bank Angle vs Time for the Sine Curve Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index 69

List of Figures (continued)

4.24: Aircraft Control Saturation (a) and Two Options for Command Limiting (b and c) 70

4.25: Illustration of Right (a) and Up (b) Priority Directions 71

4.26: Generation of Normal Acceleration and Bank Angle Commands With Command Limiting 72

List of Tables

A.1: Reference Frame X-Y Coordinates for the Half-Circle Waypoint Set 79

A.2: Reference Frame X-Y Coordinates for the Sine Curve Waypoint Set 80

Glossary of Symbols

$\underline{1}_x, \underline{1}_y, \underline{1}_z$	Unit Vectors Describing the Earth Reference Frame
$\underline{1}_v, \underline{1}_r, \underline{1}_u$	Unit Vectors Describing the Aircraft Frame
a_n	Aircraft Normal Acceleration (m/s^2)
a_{nc}	Commanded Aircraft Normal Acceleration (m/s^2)
$\underline{1}_a$	Unit Vector Describing a_{nc} in the $\underline{1}_r$-$\underline{1}_u$ Plane Without Command Limiting
$\underline{1}_\phi$	Unit Vector Describing a_{nc} in the $\underline{1}_r$-$\underline{1}_u$ Plane With Command Limiting
a_{rc}, a_{uc}	Command Components of a_{nc} in Aircraft Frame (m/s^2)
$\tilde{a}_{rc}, \tilde{a}_{uc}$	Components of a_{rc} and a_{uc}, Respectively, Based Upon Aircraft Position Error and Error Rate (m/s^2)
a_{rn}, a_{un}	Components of a_{rc} and a_{uc}, Respectively, Based Upon the Nominal Acceleration of the Desired Trajectory (m/s^2)
ϕ	Aircraft Bank Angle (deg)
ϕ_c	Commanded Aircraft Bank Angle (deg)
φ	Aircraft Yaw Angle (rad)
θ	Aircraft Pitch Angle (rad)
$\underline{r}_a$	Aircraft Position Vector
x_a, y_a, z_a	Components of $\underline{r}_a$ in Earth Reference Frame (m)
X_i, Y_i, Z_i	Coordinates of i^{th} Waypoint (m)
$\underline{V}_a$	Aircraft Velocity Vector
v_x, v_y, v_z	Components of $\underline{V}_a$ (m/s)
$\underline{e}$	Aircraft Tracking Error Vector
$\underline{T}$	Tangent Vector on Desired Trajectory
e_r, e_u	Components of $\underline{e}$ in the Aircraft Frame (m)
$\dot{\underline{e}}$	Rate of Change of Aircraft Tracking Error Vector

Glossary of Symbols (continued)

$\dot{e}_r, \dot{e}_u$	Components of $\dot{\underline{e}}$ in the Aircraft Frame (m/s)
g	Gravity Magnitude (9.81 m/s^2)
K_1	Position Feedback Gain for Acceleration Command (s^{-2})
K_2	Velocity Feedback Gain for Acceleration Command (s^{-1})
τ	Time Constant in Aircraft FCS Model (s)
s	Laplace Transform Variable (s^{-1})
S	Independent Variable for Curve Fitting (m)
$F(x,y,z)$	Function Describing an Ellipsoid in the Reference Frame
$\bar{E}$	3×3 Matrix Describing the Principal Axes of an Ellipsoid in the Reference Frame
$f(\beta_r, \beta_u)$	Function Describing an Ellipse in the $\underline{1}_r$-$\underline{1}_u$ Plane
∇	The Vector Gradient
$\underline{1}_\nabla$	Priority Direction Unit Vector in the $\underline{1}_r$-$\underline{1}_u$ Plane

Chapter 1

Introduction

1.1 Nature of the Problem

Current design trends are resulting in vehicular systems that place very demanding loads on vehicle pilots. As an example, many modern fighter aircraft operate in a Terrain Following / Terrain Avoidance (TF/TA) mode where the pilot is operating the aircraft very close to the ground at high speeds while trying to meet mission objectives and avoid enemy attack. As a result of these demands, research efforts are being focused in the area of computer-based systems which will provide semi-autonomous control for manned vehicles or autonomous control for unmanned vehicles. These vehicles might operate in land, sea, air, or space environments.

A schematic diagram of how a system of this nature can provide control for a vehicle is shown in Figure 1.1.

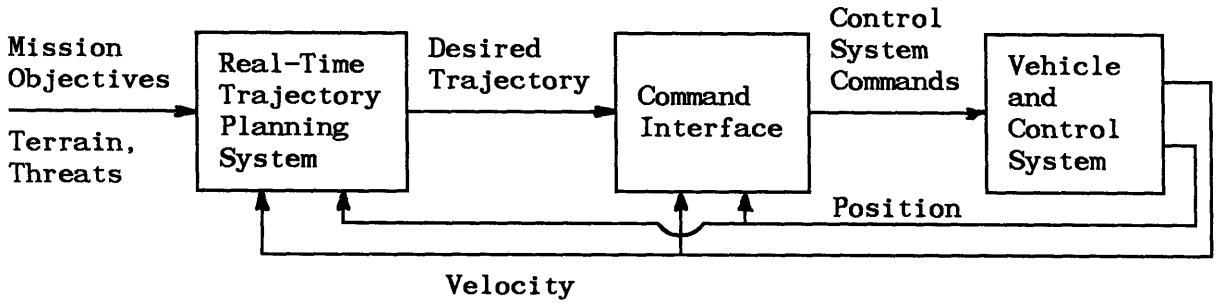

Figure 1.1 Schematic of a Computer-Based Vehicle Controller

The key element in this system is the trajectory planner. The planner computes trajectories for the vehicle based upon desired mission objectives, threats, terrain constraints, and other important factors. Once the planner has generated a desired vehicle trajectory, the vehicle attempts to follow the trajectory using a control system. However, the trajectories generated by the planner are not in a form which can be readily interpreted by the vehicle control system. Thus, an interface must be used which links the trajectory planner with the vehicle control system.

For some vehicle applications, such as high speed aircraft, the dynamic coupling represented by the command interface is very important. In the high speed case the vehicle environment is changing significantly in the time required for the trajectory planner to generate a new trajectory. In addition, any lags in vehicle response due to delays in the system might result in unacceptable tracking performance. Thus, the feasibility of a computer-based vehicle controller is greatly influenced by the functions and operational parameters of the command interface.

1.2 Research Objectives

The objectives of this research are focused on the study of a command interface between a trajectory planner and a vehicle control system. In particular, the main objective of this research is to investigate a command interface which links an existing trajectory planner to a conventional aircraft Flight Control System (FCS). The requirements and functions of the interface are developed and a suitable interface is devised. Additionally, the development of this command interface is done while concentrating on practical implementation issues, such as the availability

and quality of any required measured variables and the implied computational burden. The total combined system is simulated to test feasibility and performance. The results are useful for overall evaluation of such computer-based trajectory planning systems and will be necessary for implementation in an operational system.

1.3 Thesis Organization

In order to meet the desired objectives, background information must be gathered and certain subsystem models need to be developed. Chapter 2 begins by discussing the parameters and functions of a trajectory planner under development at the Charles Stark Draper Laboratory. Factors governing the choice of a suitable aircraft FCS are presented and a model for the FCS is developed.

Once these areas have been addressed, Chapter 3 discusses the overall form and functions of the aircraft command interface. Included in this section is an explanation of the physical relationship between the aircraft and its desired trajectory and the development of a simulation for testing the combined system performance.

Chapter 4 presents the simulation results and discusses the parameter choices involved in the command interface design. Conclusions of this research and recommendations for future work are given at the end of this thesis in Chapter 5.

Chapter 2

Trajectory Planner and Aircraft Flight Control System Models

2.1 The Trajectory Planner

The real-time trajectory planning system which provides the basis for this work is currently being developed at the Charles Stark Draper Laboratory. This system is hierarchical in nature and has four levels. At the highest level in the hierarchy is the goalpoint planner. The goalpoint planner generates a far-term mission plan consisting of an ordered sequence of mission goals and their geographic locations along with intergoal constraints for time and energy. The second level in the system is the waypoint planner. The waypoint planner produces maximum survivability flight path routes between goals. The third level is the modal planner which produces the ordered sequence of aircraft subsystem actions and associated flight modes spanning a sequence of two or three segments of the waypoint planner. The lowest level of this system is the trajectory planner. The trajectory planner creates trajectories for the vehicle along segments of the modal planner. These trajectories are based upon a database of terrain, known threats, desired mission objectives, and other factors. They are described in space by discrete waypoints that are located at the nodes of a three dimensional grid with 100 meter divisions. The desired trajectory is then given by straight line segments which

connect these waypoints. Thus, the command interface must link this lowest level of the trajectory planning system with the aircraft FCS.

To create a simulation of this trajectory planning system, an aircraft operation mode had to be chosen. The baseline mode used is the TF/TA mode. In this operational state, the aircraft is assumed to have a constant energy based upon an initial aircraft speed of Mach 0.7 or about 240 meters/second. The model for the trajectory planner is then a list of waypoints which the aircraft is to follow. For the purposes of simulation, a static set of waypoints is used. Each waypoint set is generated by creating arbitrary discrete trajectories such as straight lines or circular turns which meet the following constraint: the curvature of the trajectories must be sufficiently large so that at Mach 0.7 the aircraft normal acceleration would not exceed a given limit. In the TF/TA mode, such a limit is usually imposed by aircraft structural limitations; consequently, a value of four times the acceleration of gravity (4 g's) was chosen as a reasonable limit for the aircraft normal acceleration. Several static sets of waypoints were used for the simulations and they will be discussed in detail in later sections.

It is important to note that the desired trajectories generated by the trajectory planner are implied to be straight line segments joining the waypoints. This type of trajectory has sharp corners at the waypoints. The aircraft can not possibly turn sharp corners of this nature, nor would it be desirable to do so. Thus, various methods were considered for creating a smooth trajectory for tracking purposes. The decision was made to place the smoothing function in the command interface and not in the trajectory planner. The actual smoothing of the waypoints and the associated parameters will be discussed later in Chapter 3.

2.2 Aircraft Flight Control System Model

In order to create a model for the aircraft FCS, the first area which must be addressed is the choice of command variables. This choice has a direct effect on the dynamic interaction between the trajectory planner and the aircraft. Also, the choice of command variables will determine the complexity and response characteristics of the FCS. In order to carry out an efficient evaluation of the overall system performance, the model of the FCS should be as simple as possible while maintaining an accurate representation of an actual system.

As mentioned in the previous section, the aircraft is operating in a TF/TA mode with constant energy. Thus, it will not be necessary for the FCS model to include control of the aircraft thrust or the resulting velocity. In this case, the aircraft motion can be controlled by commanding forces normal to the velocity vector. In order to achieve this, the FCS has several command variable options. For the longitudinal channel, a few choices include pitch attitude, pitch rate, angle of attack, or normal acceleration magnitude. Some of the command variable alternatives in the lateral channel include roll rate, yaw rate, and bank angle. The most straightforward combination is a two channel FCS that controls normal acceleration magnitude in the longitudinal channel and bank angle in the lateral channel. This two channel system was chosen for the aircraft FCS model.

To describe the FCS dynamics, each channel was assumed to be a first-order lag as shown in Figure 2.1. The normal acceleration and bank angle commands are given by a_{nc} and ϕ_c, respectively. The actual values are equal to the commanded values after the first-order lag in each channel and

are given by a_n and ϕ. A time constant of $\tau = 0.25$ seconds was used in each channel model. This value is considered to be realistic since, for each first-order system, the actual values will be within 5% of the commanded values after 3 time constants, or 0.75 seconds. A response time of 0.75 seconds might actually be conservative when considering the performance characteristics of modern fly-by-wire aircraft systems, but it is in accordance with previous results of F-4 fighter aircraft simulations done at the Charles Stark Draper Laboratory [1].

The FCS model presented above is simple, but it is quite adequate for the purposes of simulating the combined planner-interface-FCS system and testing the command interface performance.

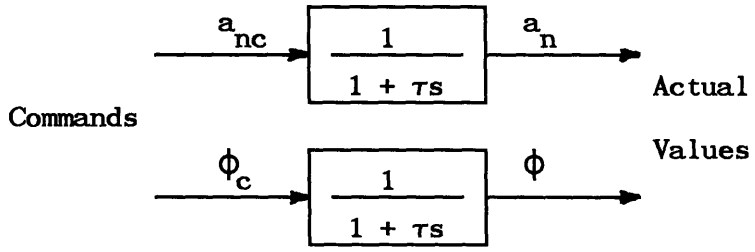

Figure 2.1 Model for Each Channel of the Aircraft FCS

Chapter 3

The Aircraft Command Interface

3.1 Introduction

In order to develop the aircraft command interface, appropriate notation and coordinate frames need to be developed. A diagram illustrating some of the notation used is shown in Figure 3.1. Recall from Chapter 2 that the command interface will have the task of creating a smooth trajectory from the waypoints for tracking purposes. To simplify the current discussion, assume that the interface has created a smooth trajectory as shown in Figure 3.1.

To begin, assume that the aircraft is equipped with an inertial navigation system that can provide the interface with the following aircraft states: position vector ($\underline{r}_a$), velocity vector ($\underline{V}_a$), normal acceleration magnitude (a_n), and bank angle (ϕ). The $\underline{r}_a$ and $\underline{V}_a$ vectors are measured relative to an Earth reference frame, which is the same frame used by the trajectory planner to generate the waypoints. In this reference frame, the Earth is assumed to be flat with the X-axis ($\underline{1}_x$) and Y-axis ($\underline{1}_y$) forming the ground plane and the Z-axis ($\underline{1}_z$) having positive values into the Earth.

An aircraft frame is also defined. In order to simplify the aircraft frame development, both the angle of attack and the sideslip angle were

assumed to be small so that the wind axes would coincide with the body axes. Therefore, $\underline{1}_v$, a unit vector in the direction of $\underline{V}_a$, is used as the longitudinal axis of the aircraft frame. A "right" unit direction ($\underline{1}_r$) is used as the aircraft lateral axis and is formed by taking the unit cross product of $\underline{1}_v$ with the negative Z-axis direction ($-\underline{1}_z$). To complete the aircraft frame, an "up" unit direction ($\underline{1}_u$) is formed by taking the unit cross product of $\underline{1}_r$ with $\underline{1}_v$.

These frames, and the associated notation, form the basis for the aircraft command interface. With this basis, the discussion can now move to the first task of the command interface: interpreting the waypoints from the trajectory planner.

3.2 Creating Smooth Desired Trajectories

As mentioned in section 2.1, the trajectories implied by the trajectory planner can not be physically attained by the aircraft due to their sharp corners. Additionally, recall that the aircraft trajectory must have a curvature of sufficient magnitude at all points such that the aircraft's normal acceleration does not exceed 4 g's. Assuming that the aircraft is travelling at a speed of Mach 0.7, this constraint means that the radius of curvature of the trajectory must be greater than approximately 1450 meters. With a waypoint resolution of 100 meters, it is obvious that trajectories having this curvature constraint can not pass through every waypoint. Thus, it was assumed that the aircraft should follow the general direction of the waypoints while not actually passing through each one.

One way to create a smooth trajectory from a set of waypoints is to

use a curve fitting routine of some nature. Since the desired trajectory should follow the general direction of the waypoints, a least-squares curve fitting scheme is appropriate. Two applications of least-squares fitting were used, and each will be discussed shortly. The method of least-squares curve fitting is presented in detail in [2].

The information provided by the trajectory planner is a list of desired waypoints which are coordinatized in the reference frame. In order to fit a curve through these waypoints, an independent variable is needed. Since the waypoints are not generated with respect to a time reference, time can not be used as the independent variable. Two methods were used to create an independent variable, and both were satisfactory. For a curve fit of n waypoints, the first method generated a variable S_i such that

$$S_k = k \cdot \Delta_n \tag{3.1}$$

for $k = 1,2,\ldots,n$ with

$$\Delta_n = \frac{\sqrt{(X_n - X_1)^2 + (Y_n - Y_1)^2 + (Z_n - Z_1)^2}}{n} . \tag{3.2}$$

This method was developed as an approximation to the average distance between each of the n waypoints in the set and is also computationally fast when compared to the second method. The second method created S_i such that

$$S_1 = 0 \text{ and}$$

$$S_k = S_{k-1} + \sqrt{(X_k - X_{k-1})^2 + (Y_k - Y_{k-1})^2 + (Z_k - Z_{k-1})^2} \tag{3.3}$$

for $k = 2,3,\ldots,n$.

This second method of independent variable definition was chosen over the first method since it was felt to be a more accurate representation of arc-length along the trajectory. The additional computational burden was

considered acceptable when considering the increased accuracy of this method.

With the creation of an independent variable, curve fitting routines can be developed to define functions representing each coordinate of the series of waypoints as a function of the independent variable. The two types of fitting schemes used will be discussed in the following sections. Each scheme divided a static set of waypoints into subsets of varying sizes which were fit to least-squares polynomials. Polynomials from first- to fourth-order were tested. The division of the static waypoint set into smaller subsets was done since several small curves can represent the waypoint coordinates with much higher accuracy than a single large curve could. In other words, the average fitting error in a curve fit of five waypoints is much lower than the average fitting error in a curve fit of 50 waypoints. Also, in a practical implementation setting, the command interface might only have a limited amount of future waypoint information from the trajectory planner. This would indicate that the command interface will have a limited event horizon and should therefore use smaller waypoint subsets to allow faster updating.

3.2.1 Unconstrained Least-Squares Curve Fitting

The first application of least-squares curve fits was used without any constraints on continuity between successive curve fits. Instead, the subsets were overlapped as shown in Figure 3.2. The subset size shown here is 6 waypoints, and each subset overlaps the previous one by 5 waypoints. As an example, assume that the total number of waypoints in the static simulation set is 69. The first curve fit set would be through waypoints 1 to 6, the second through waypoints 2 to 7, and so on. With this subset size of 6 waypoints, the interface would generate 64 subsets of curve fits

for each waypoint coordinate.

The intent in this fitting application was that the interface would switch the tracking from one subset to the next as the aircraft followed the trajectory. A particular curve was used as the desired trajectory only while the aircraft traversed the central part of the subset of waypoints used to generate that curve. The overlapping was done so that the aircraft's measured tracking error would not have large jumps when switching subsets. As will be shown in Chapter 4, this was not found to be true. Switching subsets that were not constrained to be continuous resulted in large spikes in the aircraft normal acceleration and bank angle time histories. This performance was deemed unsatisfactory and another application of least-squares curve fitting using constraints was tested.

3.2.2 Constrained Least-Squares Curve Fitting

In order to remedy the spiky behavior encountered with the unconstrained fits, constraints were applied to the subsets. The first attempt forced the subsets to be continuous in position and slope. An additional constraint forcing continuity in curvature produced the best simulation results.

As an example of this application, assume that third-order polynomials are being used and that the subset size is nine waypoints. The interface would begin by fitting a least-squares curve of third-order through waypoints 1 to 9; the position, slope, and curvature of this fit at the first waypoint would match that of the current aircraft position, velocity, and acceleration. Next, the interface would fit a curve through waypoints 9 to 17; the position, slope, and curvature of this fit would match those of the previous curve evaluated at the ninth waypoint. This would continue until the end of the static set of waypoints was reached.

A variation of this method was finally found to produce the best results. The position, slope, and curvature constraints were still applied, and in addition the subsets were overlapped as in the unconstrained case. In the previous example with a subset size of 9, assume an overlapping of 2 waypoints. The first curve would again be a fit of waypoints 1 to 9. The second curve, however, would be a fit of waypoints 7 to 15; the position, slope, and curvature of the second curve would begin with the position, slope, and curvature values of the first curve evaluated at the seventh waypoint. The interface would use the first curve as the desired trajectory until the aircraft passed the seventh waypoint and would then switch to the second curve.

This overlapping was done at different indices in the subset and thus provided another parameter in the curve fitting scheme. This will be discussed further in Chapter 4 with the simulation results.

3.3 Command Generation Algorithm

Due to the nature of the FCS model, the command generation algorithm must produce normal acceleration and bank angle commands. These commands should be generated so that the normal acceleration vector of the aircraft will cause the aircraft to track the desired trajectory. If the interface can compute the reference frame acceleration which will result in proper tracking, this acceleration can be transformed into the aircraft frame to give right and up components of required aircraft acceleration.

The structure for each right and up component of commanded aircraft acceleration has two parts. The right component of the commanded aircraft acceleration is given by

$$a_{rc} = \tilde{a}_{rc} + a_{rn} \tag{3.4}$$

and the up component of the commanded aircraft acceleration is given by

$$a_{uc} = \tilde{a}_{uc} + a_{un} \ . \tag{3.5}$$

The first terms in each command component, $\tilde{a}_{rc}$ and $\tilde{a}_{uc}$, are based upon the aircraft tracking error vector ($\underline{e}$) and the rate of change of this vector ($\dot{\underline{e}}$). The second terms, a_{rn} and a_{un}, are based upon the nominal acceleration which the aircraft would have if it were on the desired trajectory.

3.3.1 Acceleration Commands Based on Tracking Error

In order to generate commands based upon the aircraft tracking error vector and the rate of change of this vector, the vectors $\underline{e}$ and $\dot{\underline{e}}$ need to be interpreted in the aircraft frame. Referring back to Figure 3.1, $\underline{r}_{min}$ is the position on the desired trajectory such that the tracking error vector $\underline{e}$ is perpendicular to the tangent vector $\underline{T}_{min}$ at $\underline{r}_{min}$. In the reference frame the tracking error is then given by

$$\underline{e} = \underline{r}_{min} - \underline{r}_a \tag{3.6}$$

and the rate of change of this error is given by

$$\dot{\underline{e}} = |\underline{V}_a| \cdot \underline{T}_{min} - \underline{V}_a . \tag{3.7}$$

Using inner product relations, the right (subscript r) and up (subscript u) components of $\underline{e}$ and $\dot{\underline{e}}$ in the aircraft frame are:

$$e_r = \underline{e} \cdot \underline{1}_r \tag{3.8}$$

$$\dot{e}_r = \dot{\underline{e}} \cdot \underline{1}_r \tag{3.9}$$

$$e_u = \underline{e} \cdot \underline{1}_u \tag{3.10}$$

$$\dot{e}_u = \dot{\underline{e}} \cdot \underline{1}_u \tag{3.11}$$

Note that $\dot{e}_r$ and $\dot{e}_u$ are not the time derivatives of e_r and e_u. With the values from (3.8) to (3.11), the portions of the right and up commanded accelerations based upon $\underline{e}$ and $\dot{\underline{e}}$ can be found. The right component is

given by

$$\tilde{a}_{rc} = K_1 e_r + K_2 \dot{e}_r \tag{3.12}$$

and the up component is given by

$$\tilde{a}_{uc} = K_1 e_u + K_2 \dot{e}_u - g (\underline{1}_u \cdot \underline{1}_z). \tag{3.13}$$

The scalar gains K_1 and K_2 were found by modelling the FCS as one that commands acceleration through a first-order lag plant with two free integrators providing feedback of velocity and position. This model is shown in Figure 3.3. The baseline values for the model of Figure 3.3 are

$$\begin{aligned} \tau &= 0.25 \text{ s} \\ K_1 &= 1.0 \text{ s}^{-2} \\ K_2 &= 1.5 \text{ s}^{-1} \end{aligned} \tag{3.14}$$

These values give the model in Figure 3.3 a damping ratio of 0.707 and were found to give the best response in the simulations.

3.3.2 Acceleration Commands Based on Nominal Acceleration

If the command generation algorithm consisted of just the $\tilde{a}_{rc}$ and $\tilde{a}_{uc}$ components, the only trajectory which the aircraft could fly with zero tracking error is a straight line. For more general trajectories, the aircraft would not be able to track the desired trajectory without having a forced tracking error. In order to improve this performance, additional feedforward terms, a_{rn} and a_{un}, are added to (3.12) and (3.13) as shown in (3.4) and (3.5). These terms cause the interface to continue command generation in the absence of tracking errors.

The calculation of the feedforward terms is based upon the form of the desired trajectory. The subscript n refers to nominal acceleration, since the components a_{rn} and a_{un} are determined by calculating the nominal normal acceleration which the aircraft would have if it were on the desired trajectory. To do these calculations, recall that the form of the desired

trajectory is a least-squares curve fitting of the waypoints done with respect to an independent variable S. The variable S can be assumed to represent arc length along the trajectory. If this assumption is made, kinematic properties of the desired trajectory can be developed which will allow the calculation of the nominal acceleration at any point along the desired trajectory.

The development of these kinematic properties is given in [3]. For general three-dimensional curvilinear motion along a smooth curve, it is possible to define a set of unit vectors at all points along the curve centered at the body with the orientation of these vectors determined by the motion. This is shown in Figure 3.4. The unit vectors are:

$$\begin{aligned} \underline{1}_t &= \text{unit tangent vector} \\ \underline{1}_n &= \text{unit normal vector} \\ \underline{1}_b &= \underline{1}_t \times \underline{1}_n = \text{unit binormal vector} \end{aligned} \tag{3.15}$$

The vectors $\underline{1}_t$ and $\underline{1}_n$ define the osculating plane or the plane of curvature. In this plane, the curvature, κ, is defined as the rate at which the angular orientation of $\underline{1}_t$ changes with respect to the path length, S. Thus,

$$\frac{d\underline{1}_t}{dS} = \kappa \, \underline{1}_n = \rho^{-1} \, \underline{1}_n \tag{3.16}$$

where ρ is the radius of curvature and is the reciprocal of κ. In the case of constant energy flight along this curve, the velocity has the form

$$\underline{v} = \frac{dS}{dt} \underline{1}_t = |\underline{v}| \, \underline{1}_t = v \, \underline{1}_t \tag{3.17}$$

and the nominal acceleration is given by

$$\underline{a} = \frac{d\underline{v}}{dt} = \dot{v} \, \underline{1}_t + v^2 \, \kappa \, \underline{1}_n \, . \tag{3.18}$$

As mentioned in section 2.1, the aircraft is in a constant energy mode and the aircraft motion can be controlled by commanding the aircraft normal

acceleration. In this case, the nominal tangential acceleration component in (3.18) can be ignored as far as command generation is concerned. Thus, if the aircraft was on the desired trajectory at a point p, the nominal normal acceleration of the aircraft should be

$$\underline{a}_n = v^2 \kappa \underline{1}_n \tag{3.19}$$

where v is the nominal aircraft speed at point p with κ and $\underline{1}_n$ evaluated on the desired trajectory at point p.

In order to use these properties, assume that the aircraft position is again as in Figure 3.1. Let x_{min},y_{min},and z_{min} be the reference frame coordinates of the point on the desired trajectory where $\underline{e}$, $\underline{r}_{min}$, and $\underline{T}_{min}$ are calculated. The calculation of a_{rn} and a_{un} would proceed in the following way. The nominal acceleration is the acceleration which the aircraft would have if it were at the point x_{min},y_{min},z_{min}. If this were true, then $\underline{1}_t$ could be calculated by making $\underline{T}_{min}$ a unit vector. With this analytic expression for $\underline{1}_t$ as a function of S, $\underline{1}_n$ and κ could be calculated by component differentiation of $\underline{1}_t$ with respect to S. In order to get the nominal aircraft speed, recall the constant energy relationship

$$E = 0.5\, v^2 + g\, h \tag{3.20}$$

where h is the altitude of the aircraft. Since E is calculated based upon the initial aircraft speed and altitude, constant energy allows us to find the speed given the altitude. Thus, the nominal speed is found by solving (3.20) for v with h set to the value of $-z_{min}$. With the nominal speed and curvature, the nominal aircraft normal acceleration $\underline{a}_n$ along $\underline{1}_n$ is calculated from (3.19).

In order to obtain values for a_{rn} and a_{un}, a nominal aircraft frame must be formed. If the aircraft were on the desired trajectory, then $\underline{V}_a$ would be along $\underline{1}_t$. Thus, the nominal right and up aircraft axes are given

by

$$\underline{1}_{rn} = \text{UNIT} \left[\underline{1}_t \times -\underline{1}_z \right] \tag{3.21}$$

$$\underline{1}_{un} = \underline{1}_{rn} \times \underline{1}_t \tag{3.22}$$

and the nominal right and up acceleration commands are

$$a_{rn} = \underline{a}_n \cdot \underline{1}_{rn} \tag{3.23}$$

$$a_{un} = \underline{a}_n \cdot \underline{1}_{un} \tag{3.24}$$

The use of these nominal acceleration commands results in better tracking performance since commands are still generated in the absence of tracking errors.

3.3.3 Normal Acceleration and Bank Angle Commands

With the values for the right and up acceleration commands as given by (3.4) and (3.5), the normal acceleration and bank angle commands for the FCS can be computed. This is shown in Figure 3.5. The normal acceleration command (a_{nc}) is the vector sum of the right and up components, and the bank angle command (ϕ_c) is the angle between the normal acceleration command vector and the $\underline{1}_u$ axis.

This form of command generation has a characteristic that is very important to recognize. Since the aircraft normal acceleration can assume both positive and negative values, commands generated as in Figure 3.5 can be realized in two ways. The first realization would be a positive normal acceleration at a given bank angle and the second one would be with a negative normal acceleration at 180 degrees from the given bank angle. This characteristic caused some problems in the simulation and will be discussed in more detail in Chapter 4.

3.4 Simulating the System Dynamics

A schematic of the command interface simulation model is shown in Figure 3.6. The system dynamics for this process are described by eight first-order differential equations of the following form:

$$\dot{\underline{r}}_a = \underline{V}_a \tag{3.25}$$

$$\dot{\underline{V}}_a = \underline{a} \tag{3.26}$$

$$\dot{a}_n = 1/\tau \left[a_{nc} - a_n \right] \tag{3.27}$$

$$\dot{\phi} = 1/\tau \left[\phi_c - \phi \right] \tag{3.28}$$

The aircraft dynamics are described by (3.25) and (3.26), while the dynamics of the FCS are represented by (3.27) and (3.28). The aircraft normal acceleration can be transformed from the aircraft frame into the reference frame using the Euler coordinate transformation as in [4]

$$\begin{aligned} a_x &= a_n \left(-\cos\phi \sin\theta \cos\psi - \sin\phi \sin\psi \right) \\ a_y &= a_n \left(-\cos\phi \sin\theta \sin\psi + \sin\phi \cos\psi \right) \\ a_z &= -a_n \cos\phi \cos\theta + g \end{aligned} \tag{3.29}$$

where the Euler angles θ and ψ are given by

$$\theta = \sin^{-1} \left[-v_z / |\underline{V}_a| \right] \tag{3.30}$$

$$\psi = \tan^{-1} \left[v_y / v_x \right] . \tag{3.31}$$

A fourth-order Runge-Kutta method [2] was implemented to integrate the system dymamics (3.25) through (3.28) subject to the constraints (3.29), (3.30), and (3.31). The results of these static simulations will be discussed in detail in Chapter 4.

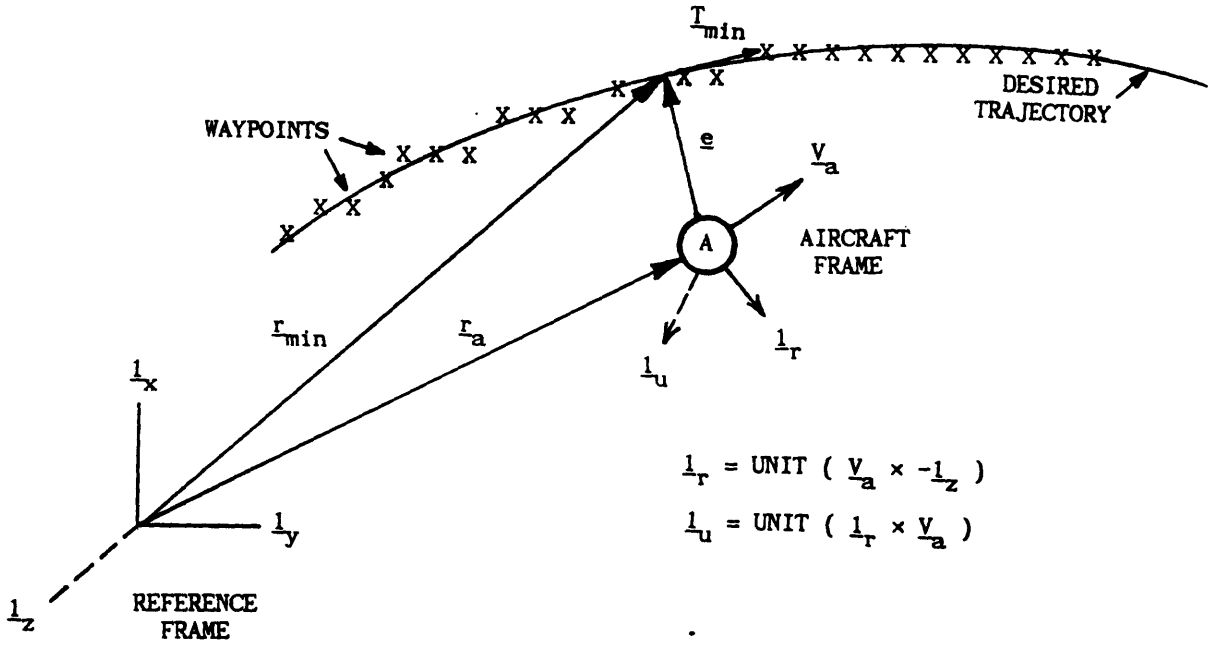

Figure 3.1 Definition of Notation

Subset 1: Fit waypoints 1-6

```
1   2   3   4   5   6   7   8   9
X---X---X---X---X---X   X   X   X
```

Subset 2: Fit waypoints 2-7

```
1   2   3   4   5   6   7   8   9
X   X---X---X---X---X---X   X   X
```

Subset 3: Fit waypoints 3-8

```
1   2   3   4   5   6   7   8   9
X   X   X---X---X---X---X---X   X
```

Figure 3.2 Subsets of Waypoints for Unconstrained Least-Squares Curve Fits

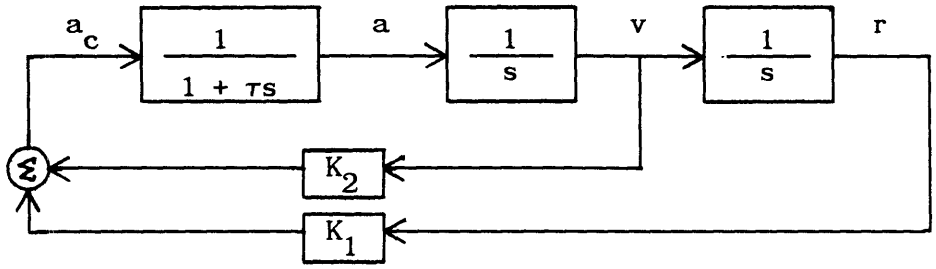

Figure 3.3 Model Used to Determine K_1 and K_2

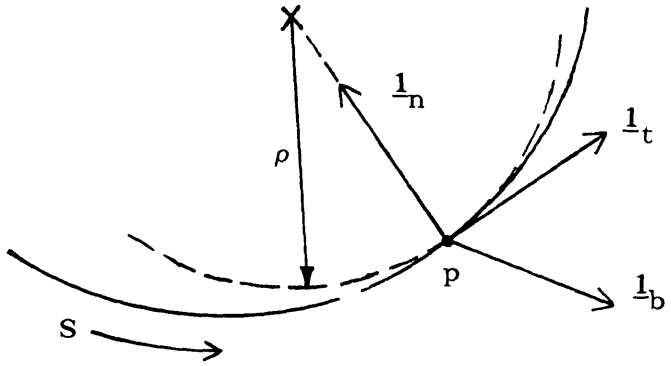

Figure 3.4 Unit Vector Orientation for General Curvilinear Motion

$$a_{nc} = \left[a_{rc}^{\,2} + a_{uc}^{\,2} \right]^{1/2}$$

$$\phi_c = \tan^{-1}\left[a_{rc} / a_{uc} \right]$$

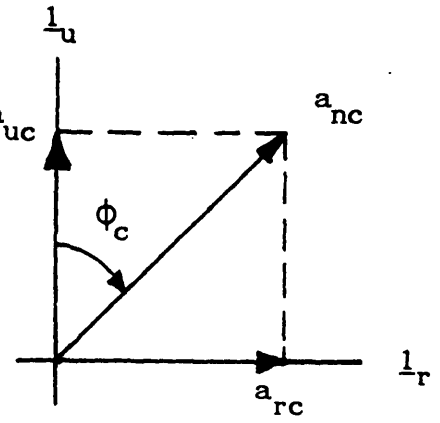

Figure 3.5 Generation of Normal Acceleration and Bank Angle Commands

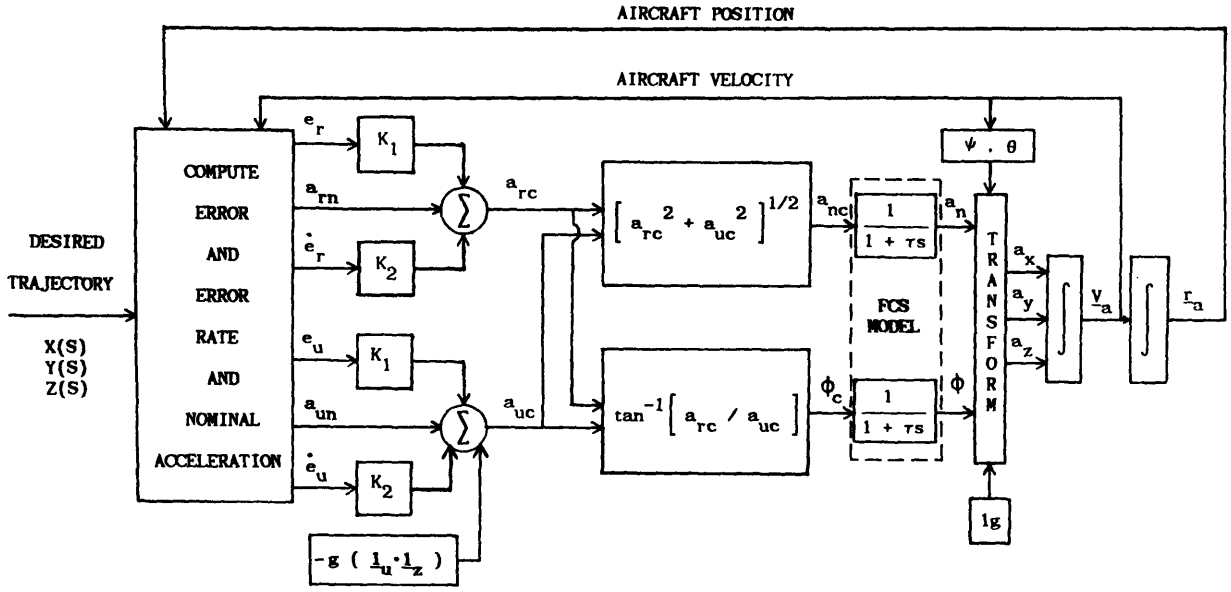

Figure 3.6 Simulation Model of Aircraft Command Interface

Chapter 4

Simulation Results

4.1 Parameter Choices in the Simulation

In order to discuss the results of the static simulations, it is necessary to briefly describe the parameters involved in the simulation. These parameters govern the overall behavior of the system model and therefore the choices for these parameters shape the final form of the command interface. All of the parameters fall into two categories: those governing the curve fitting of the static waypoint set and those governing the command generation.

The most significant impact upon the simulation results came from the choices for the parameters which controlled the least-squares curve fitting of each static waypoint set. When using unconstrained curve fits, the fitting parameters were the polynomial order and the waypoint subset size for each curve. The unconstrained subsets were always overlapped such that each subset would begin with the second index of the previous subset. In the constrained curve fits, there were four fitting parameters. These four parameters were the order of polynomial, the waypoint subset size for each curve, the level of applied constraint (i.e. position, slope, and curvature), and the index in the waypoint subset where the constraint was applied. The index where the constraint is applied is called the matching

point.

For unconstrained curve fitting, sample fitting parameter values could be the use of second-order polynomials in subsets of 6 waypoints. A static set of 50 waypoints would result in 45 curve fits with indices 1-6, 2-7, and so on up to 45-50. The desired trajectory would be represented by the middle segments of each curve as shown in Figure 4.1. The interface would use the first curve as the desired trajectory until the aircraft passed the fourth waypoint. The second curve would then be tracked until the aircraft passed the fifth waypoint, and so on.

For constrained curve fitting, sample fitting parameter values could be the use of third-order polynomials in subsets of 9 waypoints which are matched in position and slope at the second index in each subset. With these values, the simulation would fit waypoints 1-9 with the initial position and slope equal to the initial position and velocity of the aircraft. Next, the second curve would be formed using waypoints 2-10 with the initial position and slope equal to the position and slope of the first curve evaluated at the second waypoint. This would continue until the last curve ended with the last waypoint in the static set. With this large overlapping of subsets, each curve is only used as the desired trajectory while the aircraft is between the first and second waypoints in the subset. Thus, the interface only uses the first curve as the desired trajectory until the second waypoint is passed, and it then switches to the second curve. This continues until the last curve is reached.

The rest of the parameters in the simulation govern the command generation; these were not changed significantly throughout the process of interface performance evaluation. These parameters include the time constants used for each channel of the aircraft FCS (see Figure 2.1) and

the scalar feedback gains used to calculate the acceleration commands (see Figure 3.3). The time constants for both FCS channels were assumed to be identical. Each was assigned the value used for the time constant in the model of Figure 3.3. In general, a value of 0.25 seconds was thought to be the most realistic and was therefore used as the value for the FCS time constants in virtually all of the simulations. With a given value for τ, K_1 and K_2 were chosen so that the model of Figure 3.3 had a damping ratio of 0.707. For $\tau = 0.25$ s, values of $K_1 = 1.0\ s^{-2}$ and $K_2 = 1.5\ s^{-1}$ were used as baseline values. Perturbing these values slightly produced simulation responses which had either larger overshoots or slower response times as compared to the simulation responses for the baseline values.

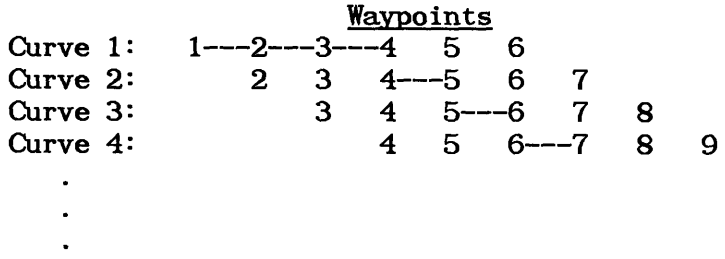

Figure 4.1 Use of Unconstrained Curve Fits to Represent the Desired Trajectory. (Desired Trajectory is Indicated by the Dashed Lines)

4.2 Unconstrained Curve Fitting Results

The use of unconstrained least-squares curve fits through the waypoints to represent the desired trajectory resulted in unsatisfactory simulation results. Several static sets of waypoints were tested. The waypoint set which best illustrates the poor results when using unconstrained fits is a right-turning half-circle of 2000 meter radius at a

constant altitude of 100 meters. The reference frame coordinates of the waypoints in this set are given in Appendix A and a plot of these coordinates in the X-Y reference plane is shown in Figure 4.2.

Sample simulation results for the half-circle waypoint set using unconstrained fits are shown in Figures 4.3 to 4.6. The unconstrained curve fitting of the waypoints was done with first-order polynomials in subsets of 6 as in Figure 4.1. Of all the various curve fitting parameter combinations used with unconstrained curve fits, this combination resulted in the smoothest time histories. With first-order fits, the nominal acceleration will always be zero since the curvature is always zero. As a result, first-order curves always produced a large average tracking error. However, when using higher order curves which have nonzero nominal acceleration, the average tracking error was less but the aircraft normal acceleration and bank angle time histories were very noisy. It was felt that the higher average tracking error produced by first-order curves was more acceptable than the time histories produced by higher order curves.

It is important to note that a constant altitude circular turn should be executed with normal acceleration and bank angle time histories that are virtually constant during the turn. However, Figures 4.5 and 4.6 show that the aircraft normal acceleration and bank angle time histories have very strong oscillations. The explanation for the oscillatory normal acceleration and bank angle time histories is found in Figure 4.4, the tracking error time history. The commands are generated from a combination of tracking error, rate of change of tracking error, and nominal acceleration of the desired trajectory. As the command interface switches from one curve to the next, it incurs jumps in the tracking error as shown in Figure 4.4. Therefore, since the tracking error is very spiky and the

nominal acceleration is zero, the commands must also be very spiky. These erratic commands cause the unacceptable aircraft normal acceleration and bank angle time histories.

4.3 Constrained Curve Fitting Results

Due to the unacceptable performance of the unconstrained curve fits, constraints were applied to the curve fitting procedure. The results were found to be acceptable for the scope of this research. As in the unconstrained case, several static waypoint sets were used to test the interface performance. In addition to the half-circle waypoint set previously mentioned, another waypoint set was extensively used. This set, which also had a constant altitude, represented one cycle of a sine curve in the X-Y reference plane.

To begin testing the constrained curve fits, the curves were forced to be continuous in position and slope. Recall that the acceleration command generation is partly based upon the tracking error and the rate of change of the tracking error. If the curve fits were forced to be continuous in position only, then the rate of change of tracking error would not be continuous when switching subsets and the commands would not be continuous when switching subsets. Thus, a constraint forcing only position continuity was not tested in the simulations.

When the curves are continuous in position and slope, the order of polynomial must be second or greater. Polynomials from second- to fourth-order were tested with various subset sizes and matching points. The best simulation results for position and slope (PS) constrained curves were obtained using second-order polynominals in subsets of nine that were

matched at the second index in each subset. Results for the half-circle waypoint set using this combination of parameters are shown in Figures 4.7 to 4.10. As compared to the unconstrained curve fitting results, it is evident that the time histories are much smoother. In addition, the nominal acceleration command components have reduced the tracking error by approximately one order of magnitude.

Even though the position and slope constraints improved the time histories, some small spikes in the aircraft normal acceleration still remained. In an attempt to make the time histories as smooth as possible, an additional constraint of curvature continuity was applied to the curve fits. With the curvature constraint, the rate of change of $\dot{\underline{e}}$ is also continuous along the trajectory and the order of polynomial must now be third or greater. Polynomials of third- and fourth-order were tested with various subset sizes and matching points. It is noted here that increasing the polynomial order from three to four never improved the results; consequently, polynomials of order greater than four were not tested. The best simulation results for position, slope, and curvature (PSC) constrained curves were obtained using third-order fits in subsets of nine that were matched at the second index in each subset. These results are shown in Figures 4.11 to 4.14. As hoped, the time histories are now smoother due to the additional curvature constraint. Note, however, that the PSC constrained results have larger overshoots in both the normal acceleration and bank angle time histories as compared to the PS constrained results. At this point, one may conclude that the slightly spiky behavior of the PS constrained case may be preferable to the larger overshoots of the PSC constrained case. However, the results for each case are quite similar and illustrate the need for further evaluation.

After fully testing the command interface using the half-circle waypoint set, the sine curve waypoint set was used. The reference frame coordinates for the waypoints of the sine curve set are given in Appendix A and a plot of these coordinates in the X-Y reference plane is shown in Figure 4.15. Note that in the half-circle waypoint set, the trajectory has an almost constant curvature during the turn. Accordingly, the PS constrained case might have been performing as if the curvature were constrained to be continuous. The sine curve waypoint set was used to represent a trajectory with varying curvature in order to see if the additional curvature constraint indeed had a pronounced effect.

As with the half-circle waypoint set, the sine curve waypoint set was tested for both PS constrained cases and PSC constrained cases using various orders of polynomials, subset sizes, and matching points. The best results for the PS constrained case (second-order polynomials, subsets of nine, matching at the second index) are shown in Figures 4.16 to 4.19. The best results for the PSC constrained case (third-order polynomials, subsets of nine, matching at the second index) are shown in Figures 4.20 to 4.23. As before, the PSC constrained case shows higher overshoots as compared to the PS constrained case. However, the spikes in the PS constrained case are quite large, especially in the aircraft bank angle time history. Comparing these results clearly shows that the additional constraint of curvature continuity is necessary to keep the aircraft normal acceleration and bank angle time histories as smooth as possible.

With the above results in mind, the curve fitting parameter choices for the final form of the command interface were: third-order fits in subsets of nine waypoints that were matched in position, slope, and curvature at the second waypoint index.

4.4 Command Limiting

The form of the command interface presented thus far results in satisfactory performance as long as the aircraft is well within its performance envelope. Even though the trajectory planner must generate trajectories which are limited in maximum curvature, some simulations resulted in aircraft normal accelerations that reached the previously established limit. As a result, the command generation algorithm needs to include logic for use during aircraft normal acceleration (a_n) saturation.

Recall from section 2.1 that the a_n limit was set to 4 g's. In general, a_n was found to range from values of 0 g to 4 g; however, in certain circumstances it was allowed to range from -1 g to 4 g. This will be discussed in the next section. For the purposes of this discussion, assume that a_n can only assume values from 0 g to 4 g.

For the two-channel FCS in the simulation, control saturation occurs when the normal acceleration command, a_{nc}, exceeds 4 g. Since a_n is limited to 4 g, any a_{nc} greater than 4 g must be scaled down. However, scaling down a_{nc} is not as straightforward as it might appear. To illustrate this, assume that the aircraft is at the altitude of the desired trajectory but has a large lateral displacement from the trajectory. Further, assume that the lateral displacement is to the "left" of the desired trajectory. Thus, the aircraft needs a large a_{rc} with a corresponding a_{uc} that is sufficient to keep the aircraft at the desired altitude. If there is no saturation and a_{nc} is greater than 4 g, the command generation will be as shown in Figure 4.24(a).

The first inclination for scaling would be to reduce a_{nc} to 4 g while keeping ϕ_c constant as shown in Figure 4.24(b). Notice that keeping ϕ_c

constant has caused a_{uc} to decrease, which will consequently cause the aircraft to lose altitude. In order to maintain altitude, a second scaling method could be used as shown in Figure 4.24(c). Using this method, the a_{nc} is reduced to 4 g and ϕ_c is also reduced so that a_{uc} is sufficient for the aircraft to maintain altitude. However, in using the second scaling method, one assumes that maintaining altitude is more important than lateral tracking performance. Therefore, the problem of scaling is one of determining the best direction, or ϕ_c, for the saturated a_{nc}.

The determination of the best ϕ_c can be done by finding an aircraft priority direction. For the purposes of this research, it was assumed that the priority direction is the direction which will move the aircraft away from the closest surrounding terrain as fast as possible. Refer to Figure 4.25(a). In this case, the priority direction is right; the aircraft can afford to lose altitude but should not move further to the left. The opposite case of an up priority direction is shown in Figure 4.25(b).

Notice in both cases that the terrain boundary can be described by a boundary ellipse which is centered on the desired trajectory. Another ellipse, called the aircraft ellipse, passes through the aircraft position and is concentric to the boundary ellipse. The priority direction is then given by the inward normal on the aircraft ellipse at the aircraft position. This method of finding priority directions was utilized in the simulation.

If the command interface is to use boundary ellipses along the desired trajectory, the trajectory planner needs to provide the interface with the necessary information. It would seem that the best division of responsibility is to have the trajectory planner compute boundary ellipses for each waypoint and pass them to the command interface rather than

transferring terrain data to the interface. Consequently, it was assumed that the interface receives ellipsoids that are centered at each waypoint, with each ellipsoid defining a volume which the aircraft should remain within. The general form for an ellipsoid centered at the i^{th} waypoint is

$$F_i(x,y,z) = (\underline{r} - \underline{x}_i)^T \bar{E}_i (\underline{r} - \underline{x}_i) = 1 \tag{4.1}$$

where $\underline{r}$ is a general position vector, $\underline{x}_i$ is the position vector for the i^{th} waypoint, and $\bar{E}_i$ is a 3x3 symmetric positive definite matrix whose eigenstructure describes the size and orientation of the ellipsoid axes in the reference frame.

Since the desired trajectory will in general be close to the waypoints, the center of each ellipsoid is considered to be on the desired trajectory. If $\underline{r}_i$ is the position vector on the desired trajectory calculated for the value of S_i, then the i^{th} ellipsoid becomes

$$F_i(x,y,z) = (\underline{r} - \underline{r}_i)^T \bar{E}_i (\underline{r} - \underline{r}_i) = 1. \tag{4.2}$$

When the aircraft is between waypoints, an ellipsoid centered at the closest trajectory position can be formed by taking a combination of the two ellipsoids which the aircraft is between. As an example, assume the aircraft is between the first and second waypoints in the set. Let the closest trajectory position be given by $\underline{r}_{min}$. If d_1 is the distance from $\underline{r}_1$ to $\underline{r}_{min}$ and d_2 is the distance from $\underline{r}_{min}$ to $\underline{r}_2$, then the ellipsoid describing the current aircraft surroundings is given by

$$F_{min}(x,y,z) = (\underline{r} - \underline{r}_{min})^T \bar{E}_{min} (\underline{r} - \underline{r}_{min}) = 1 \tag{4.3}$$

where

$$\bar{E}_{min} = (1 - \alpha) \bar{E}_1 + \alpha \bar{E}_2 \tag{4.4}$$

$$\alpha = d_1 / (d_1 + d_2) . \tag{4.5}$$

The equation of the boundary ellipse is given by the intersection of the $\underline{1}_r$-$\underline{1}_u$ plane with (4.3). For any vector $\underline{r}$ in the $\underline{1}_r$-$\underline{1}_u$ plane passing

through the point $(x_{min},y_{min},z_{min})$, there exist two scalar coordinates (β_r,β_u) such that

$$\underline{r} - \underline{r}_{min} = \beta_r \underline{1}_r + \beta_u \underline{1}_u \ . \tag{4.6}$$

Substituting (4.6) into (4.3) will give the equation of the boundary ellipse. Ellipses in the $\underline{1}_r$-$\underline{1}_u$ plane are described by the quadratic

$$f_{min}(\beta_r,\beta_u) = \left[\beta_r\underline{1}_r + \beta_u\underline{1}_u \right]^T \bar{E}_{min} \left[\beta_r\underline{1}_r + \beta_u\underline{1}_u \right] \ . \tag{4.7}$$

Ellipses concentric to the boundary ellipse $f_{min}(\beta_r,\beta_u) = 1$ which pass through a given point (σ_r,σ_u) in the $\underline{1}_r$-$\underline{1}_u$ plane are described by

$$\frac{f_{min}(\beta_r,\beta_u)}{f_{min}(\sigma_r,\sigma_u)} = 1 \ . \tag{4.8}$$

Recall that e_r and e_u are the components of $\underline{e}$ in the $\underline{1}_r$-$\underline{1}_u$ plane, where $\underline{e}$ is measured from the aircraft to the desired trajectory. Therefore, with σ_r and σ_u equal to $-e_r$ and $-e_u$, respectively, (4.8) will give the equation of the aircraft ellipse as

$$f_a(\beta_r,\beta_u) = \frac{f_{min}(\beta_r,\beta_u)}{f_{min}(-e_r,-e_u)} = \frac{f_{min}(\beta_r,\beta_u)}{J} = 1 \ . \tag{4.9}$$

The scalar $J = f_{min}(-e_r,-e_u)$ determines the size of the aircraft ellipse. J ranges from zero when the aircraft is on the desired trajectory to one when the aircraft is on the boundary ellipse.

The priority direction can now be found by computing the inward normal on the aircraft ellipse at the aircraft position. The inward normal is given by the negative gradient of the function $f_a(\beta_r,\beta_u)$ and is expressed as

$$-\nabla f_a(\beta_r,\beta_u) = -\frac{\partial}{\partial\beta_r} f_a(\beta_r,\beta_u)\ \underline{1}_r - \frac{\partial}{\partial\beta_u} f_a(\beta_r,\beta_u)\ \underline{1}_u \ . \tag{4.10}$$

Therefore, the priority direction is given by the unit vector 1_∇, where

$$\underline{1}_\nabla = \mathrm{UNIT}\left[-\nabla f_a(-e_r,-e_u) \right] \ . \tag{4.11}$$

With the priority direction determined, a scaling algorithm can be

developed to generate ϕ_c so that the aircraft motion will be best suited to the aircraft surroundings.

The logic to scale saturated commands should have the property that ϕ_c remains continuous as a_{nc} varies between saturated and unsaturated values. The scaling logic developed for the simulation uses the a_{nc} generation algorithm of Chapter 3 and introduces the priority direction into the generation of ϕ_c. As before, a_{nc} is generated from the a_{rc} and a_{uc} components and is limited to a maximum value of 4 g. A unit vector in the direction of a_{nc} is given by

$$\underline{1}_a = \text{UNIT} \left[a_{rc} \underline{1}_r + a_{uc} \underline{1}_u \right] . \tag{4.12}$$

The generation of ϕ_c can be influenced by the priority direction if the normal acceleration command vector is a combination of $\underline{1}_a$ and $\underline{1}_v$. With the vector $\underline{1}_\phi$ defined as

$$\underline{1}_\phi = \text{UNIT} \left[(1 - J) \underline{1}_a + J \underline{1}_v \right] , \tag{4.13}$$

where J is given as in (4.9), ϕ_c is now defined as the angle between $\underline{1}_\phi$ and the $\underline{1}_u$ axis as shown in Figure 4.26. This form of ϕ_c generation is continuous between unsaturated and saturated a_{nc}. Note that if the aircraft is close to the desired trajectory, J will be small and the commanded bank angle will be virtually the same as previously presented. However, as the aircraft ellipse reaches the boundary ellipse, J approaches one and the command generation now directs the aircraft motion away from the surrounding terrain in the direction of $\underline{1}_v$.

This new command generation algorithm was implemented in the simulation and proved useful in dealing with control saturation. Since the tracking errors were usually on the order of 10 meters and the ellipsoids were conservatively defined with principal axes on the order of 500 meters, the actual performance did not change drastically. However, this algorithm

provides the interface with a logical framework for handling saturation and is adequate for the scope of this research.

4.5 Choice of Sign for Normal Acceleration Commands

As mentioned in section 3.3.3, allowing the aircraft normal acceleration to assume both positive and negative values creates a problem for the command generation algorithm shown in Figure 3.5. With this algorithm, there are always two command choices: one with a positive normal acceleration at a given bank angle and one with a negative normal acceleration at 180 degrees from the given bank angle. The problem lies in determining which pair of commands should be used.

For virtually all flight maneuvers, the aircraft normal acceleration command (a_{nc}) has positive values. However, if a_{nc} is always positive, large and undesirable changes in the aircraft bank angle may result. To illustrate this, visualize a ballistic trajectory at a constant heading in the X-Z plane of the reference frame. If the aircraft were on the desired trajectory, it would be travelling with constant speed in the $\underline{1}_x$ direction and freefalling in the $\underline{1}_z$ direction.

If the ballistic trajectory is flown with a_{nc} always positive, problems occur since both a_{uc} and a_{rc} are very small, and small tracking errors cause a_{uc} to switch from positive to negative values and cause ϕ_c to correspondingly switch by nearly 180 degrees. This is not desirable, since an aircraft could fly a trajectory of this nature with virtually no changes in bank angle. The aircraft performance can be made more realistic by allowing a_{nc} to assume small negative values. Aircraft and pilot constraints must be considered when determining the maximum magnitude and

duration of negative g-loading. The constraints assumed for this research were: (1) the aircraft normal acceleration magnitude can not drop below -1 g and (2) the aircraft normal acceleration can not be continuously negative for more than 10 seconds. The magnitude constraint assumption was made on a pilot comfort basis and the duration constraint assumption was made on an aircraft engine performance basis.

The command generation algorithm was updated so that the choice of commands would not violate the above constraints. The update logic is based upon the nominal acceleration along the desired trajectory. By calculating values of nominal acceleration at future points along the desired trajectory, the interface can determine what the aircraft normal acceleration will be. Criteria can then be used to determine what the current commands should be based upon the known future nominal acceleration evolution.

The development of the logic was done in the following manner. The aircraft commands were generated as previously described, with a_{nc} always positive. However, if the current ϕ_c differs from the previous ϕ_c by more than 135 degrees, the interface looks ahead to the future trajectory. Recall that the desired trajectory is described by overlapped curves such that each curve is used only when the aircraft is between the first and second waypoints in the curve subset. To calculate the future trajectory evolution, the interface computes the nominal acceleration at the next 20 subset matching points. This is equivalent to computing the nominal acceleration at the beginning of the next 20 curves. The future values of nominal right and up acceleration, a_{rn} and a_{un}, are used as a_{rc} and a_{uc} to give future values for a_{nc} and ϕ_c. This is done such that the future values of a_{nc} are always positive. The future values of ϕ_c are then

compared to the previous ϕ_c. The current a_{nc} value is kept positive (and the large change in bank angle is initiated) only if one of the following criteria is met: (1) the value of ϕ_c for all 20 of the future curves is different from the previous ϕ_c by more than 135 degrees, or (2) three consecutive curves of the 20 have a ϕ_c which differs from the previous ϕ_c by more than 135 degrees and also have an a_{nc} greater than 1 g.

The choice of using the next 20 curves for determining the future trajectory evolution stems from the baseline aircraft speed of 240 m/s. If each curve has an average length of about 100 meters, then the baseline aircraft will travel 20 curves in about 10 seconds. Thus, criterion (1) prevents the aircraft from having a negative normal acceleration for more than 10 seconds. Criterion (2) comes from the constraint that the aircraft normal acceleration can not be less than -1 g. Three consecutive curves must have a_n greater than 1 g since the aircraft could not "flip" to attain the large ϕ_c and successfully track just one or two consecutive curves. To see this, note that the FCS model time constants give the aircraft a maximum roll rate of approximately 180 deg/s; at a speed of 240 m/s, the aircraft would pass two curves before it could roll to attain a large ϕ_c.

The above logic was implemented in the simulation and tested with a ballistic trajectory and other trajectories involving climbs or dives at a constant heading. It is noted that the problem of determining the sign of a_{nc} can occur for any general trajectory where the value of a_{nc} is close to or passes through zero. The constant heading assumption was made to allow easier generation of desired trajectories which had the above a_{nc} characteristic.

This method was found to be satisfactory in dealing with sign choices for a_{nc} and is felt to be adequate for the scope of this research.

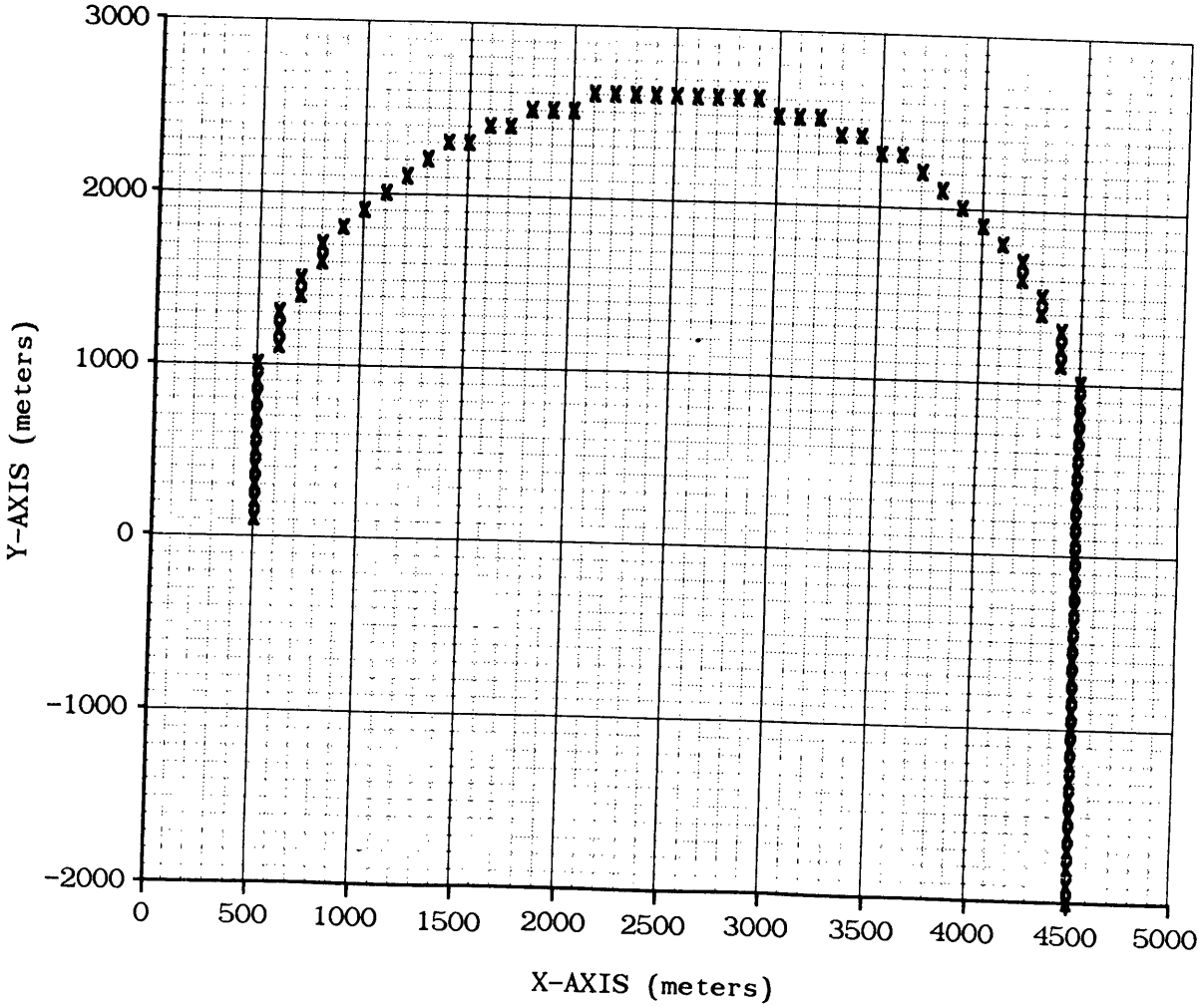

Figure 4.2 X-Y Reference Frame Coordinates for the Half-Circle Waypoint Set

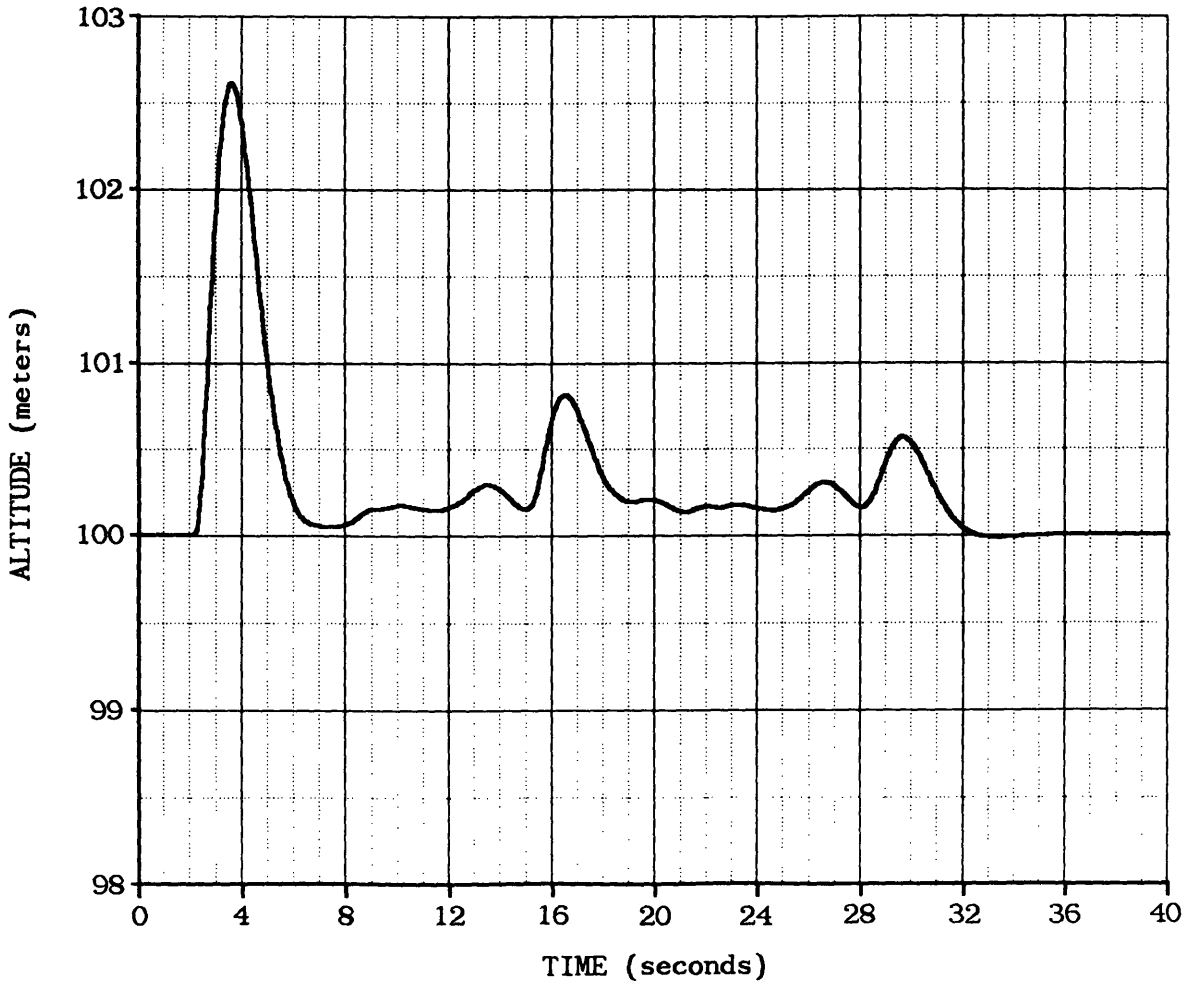

Figure 4.3 Aircraft Altitude vs Time for the Half-Circle Waypoint Set Using Unconstrained First-Order Curve Fits in Subsets of Six

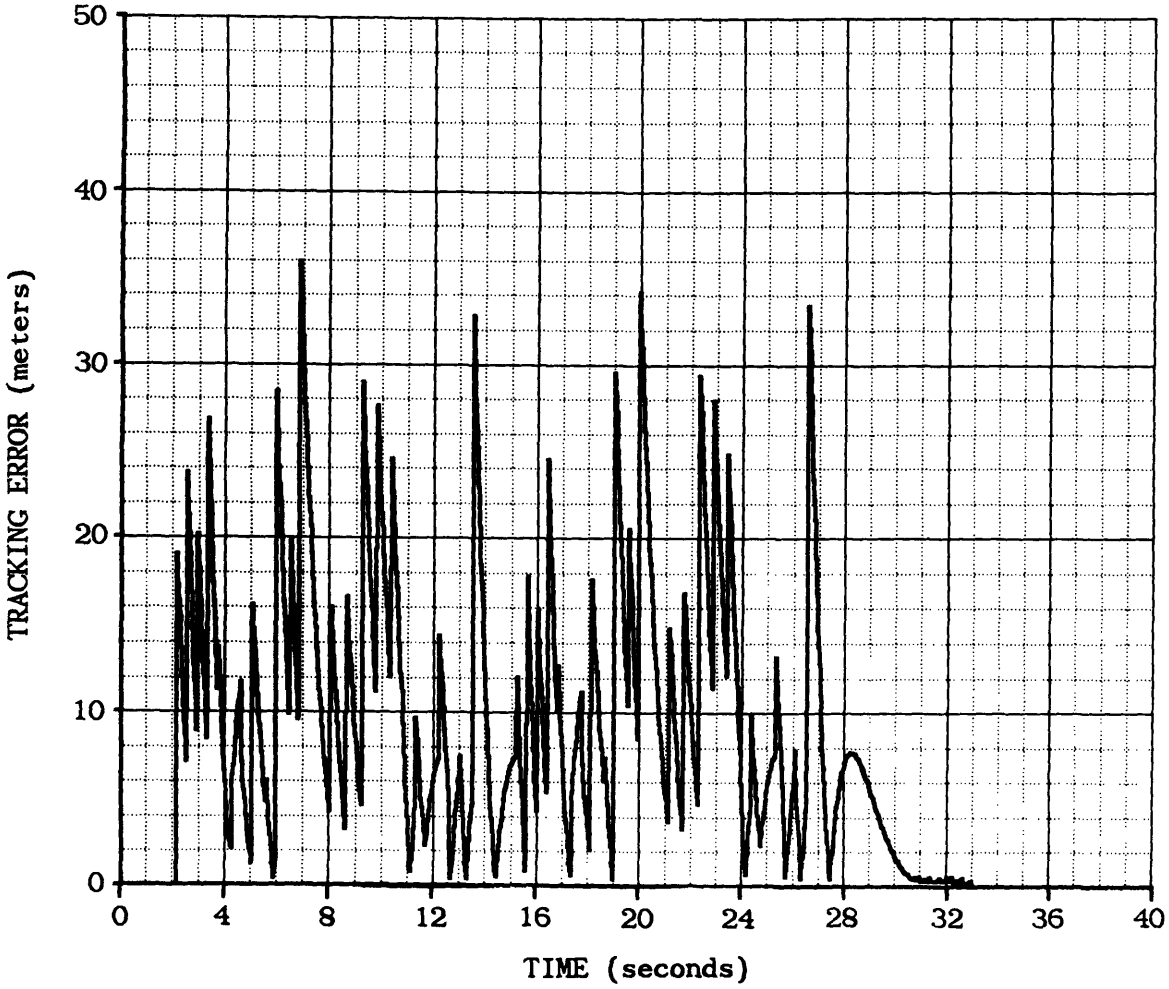

Figure 4.4 Aircraft Tracking Error vs Time for the Half-Circle Waypoint Set Using Unconstrained First-Order Curve Fits in Subsets of Six

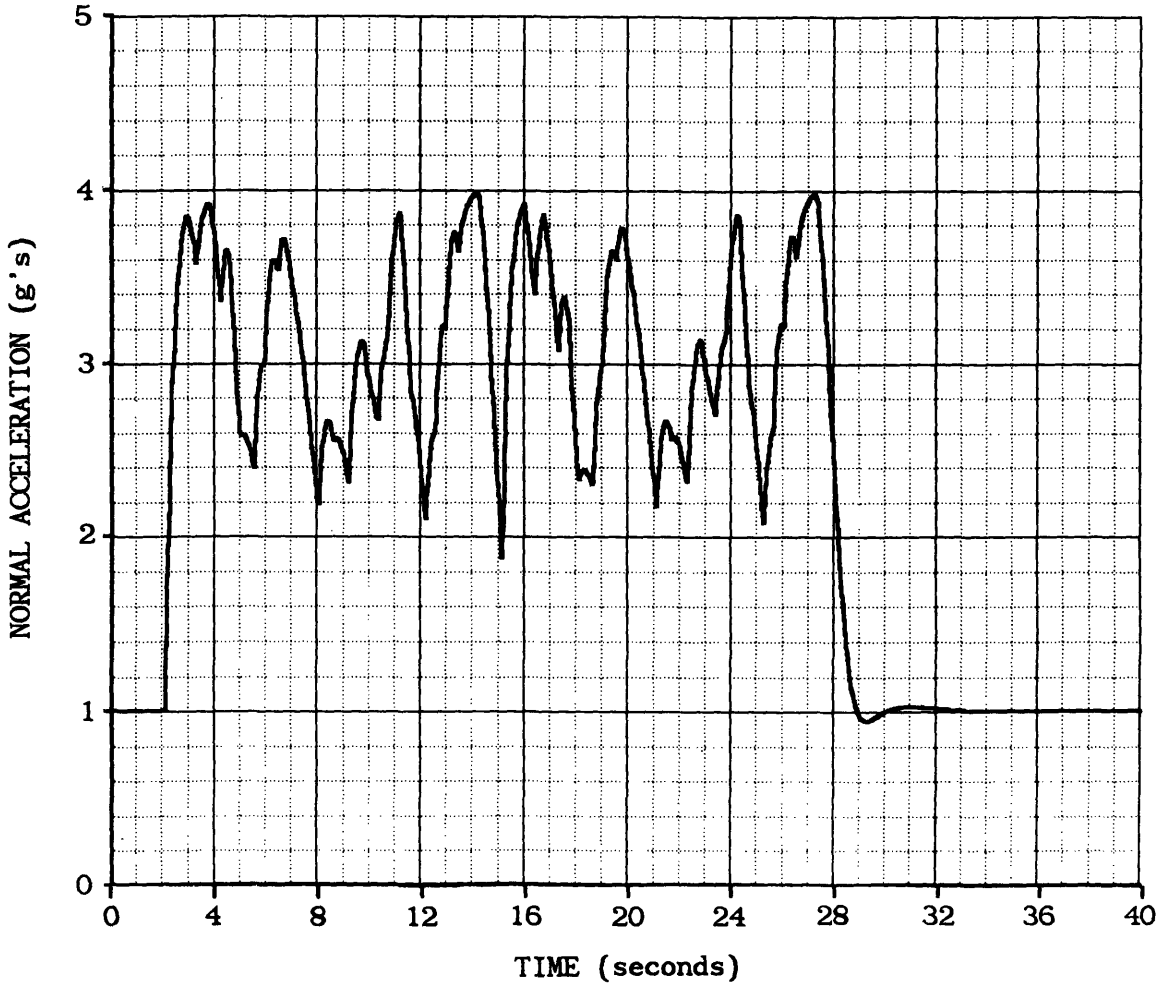

Figure 4.5 Aircraft Normal Acceleration vs Time for the Half-Circle Waypoint Set Using Unconstrained First-Order Curve Fits in Subsets of Six

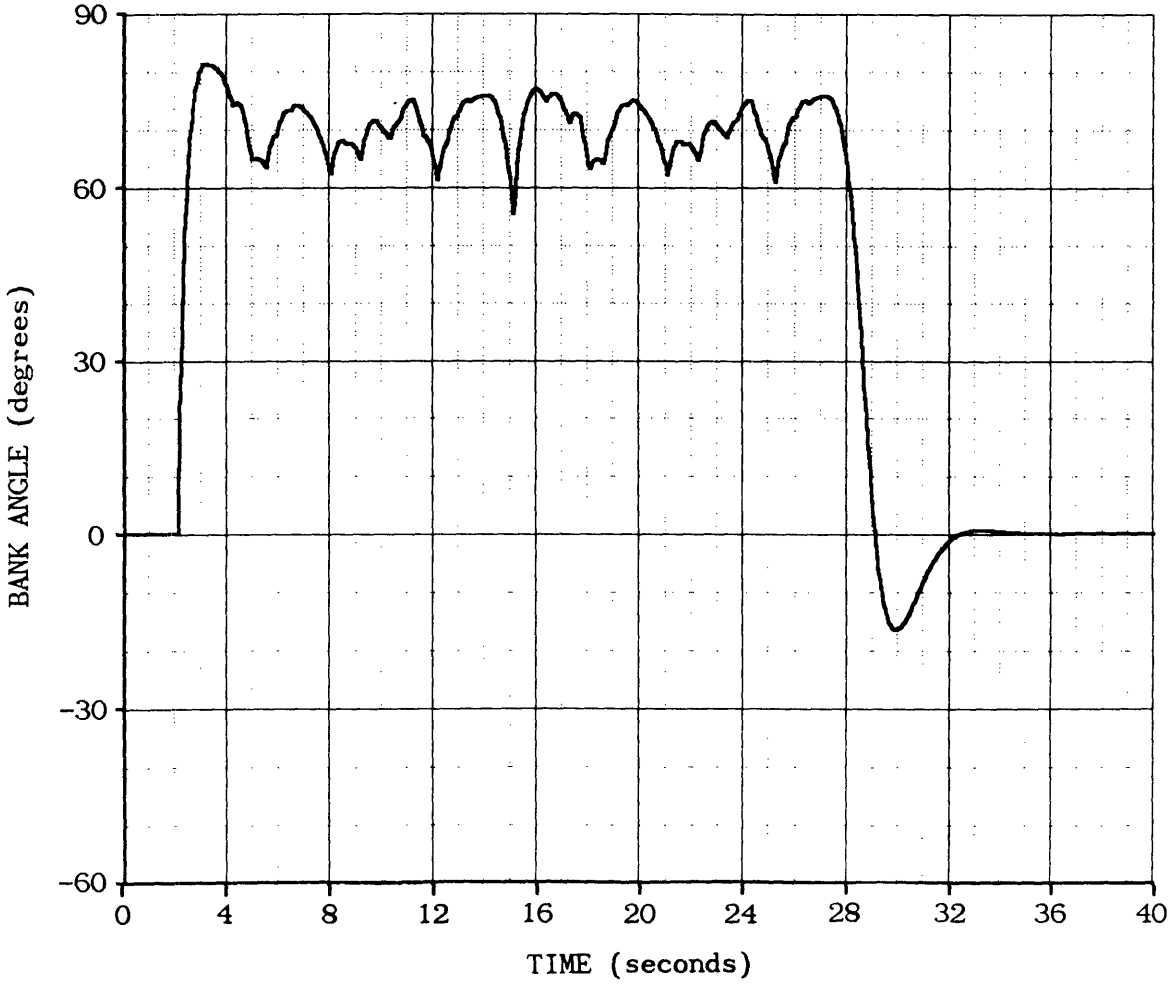

Figure 4.6 Aircraft Bank Angle vs Time for the Half-Circle Waypoint Set Using Unconstrained First-Order Curve Fits in Subsets of Six

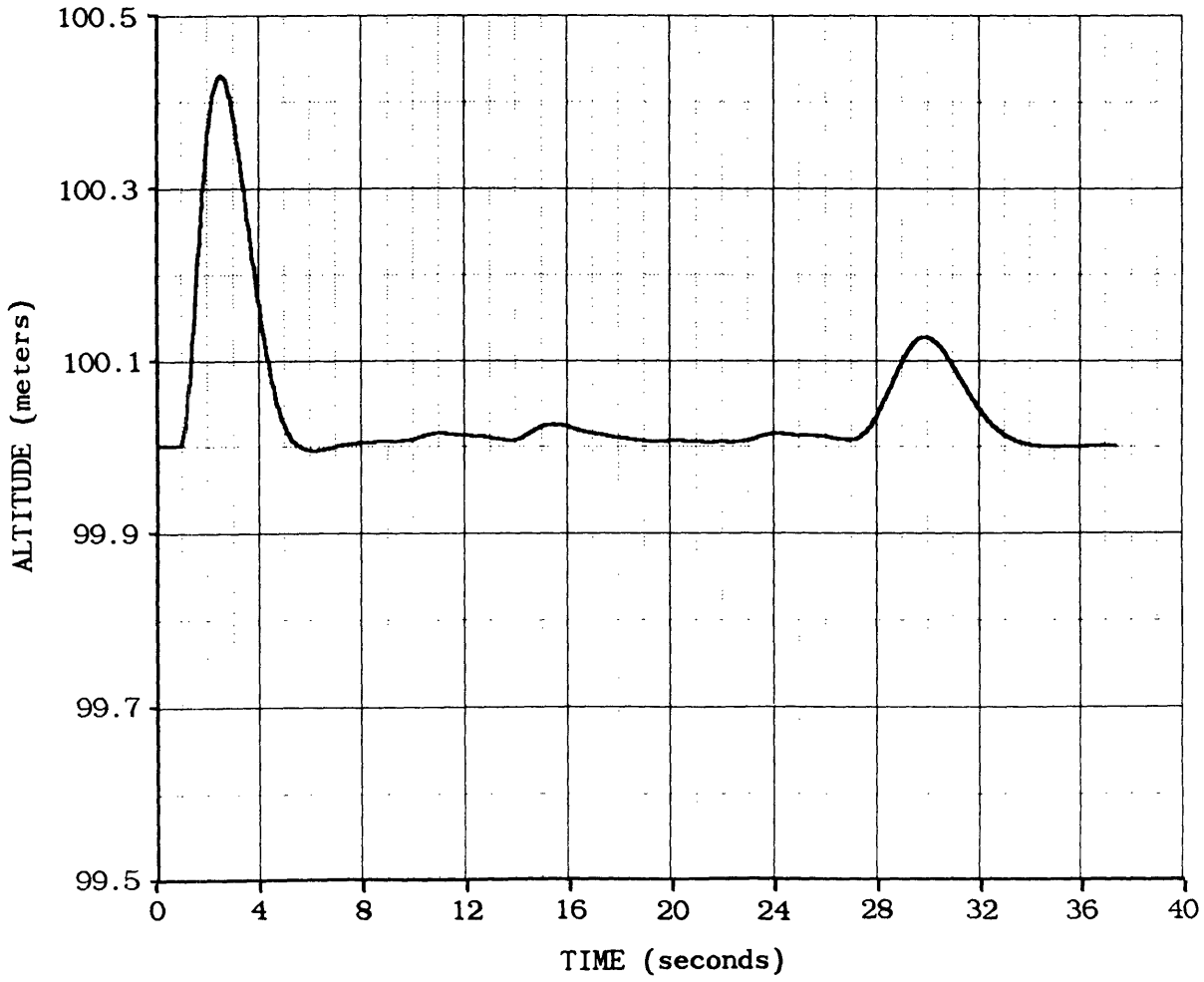

Figure 4.7 Aircraft Altitude vs Time for the Half-Circle Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index

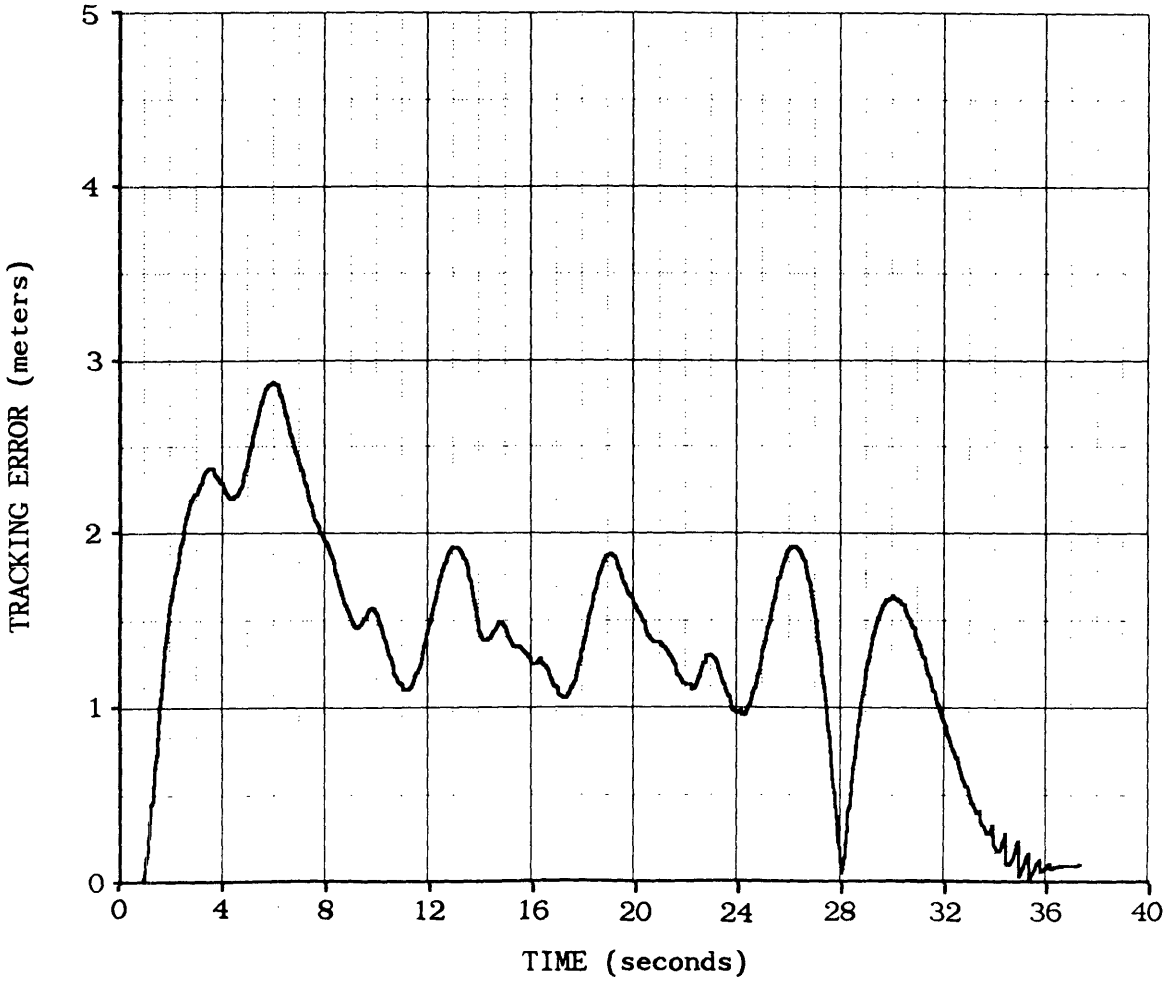

Figure 4.8 Aircraft Tracking Error vs Time for the Half-Circle Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index

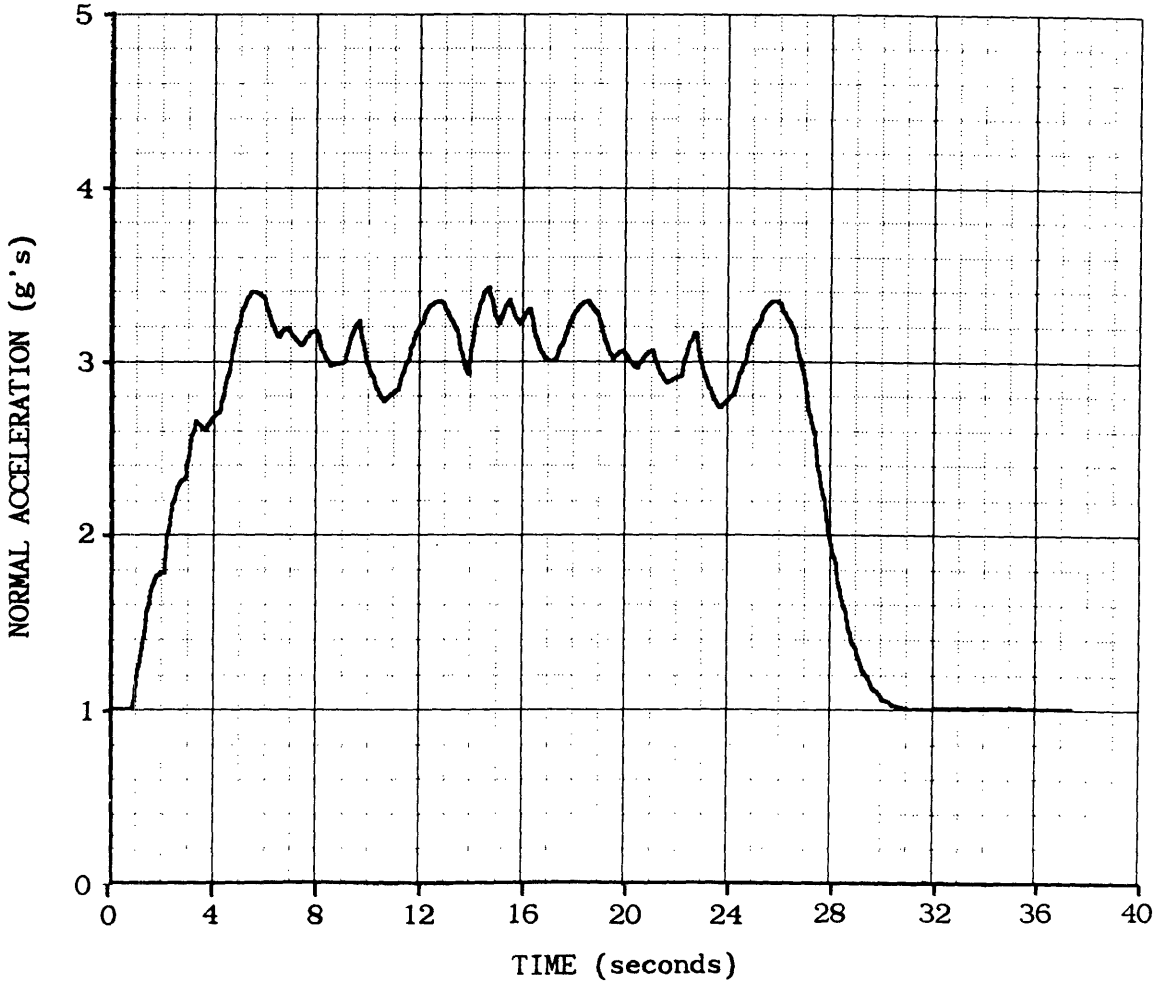

Figure 4.9 Aircraft Normal Acceleration vs Time for the Half-Circle Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index

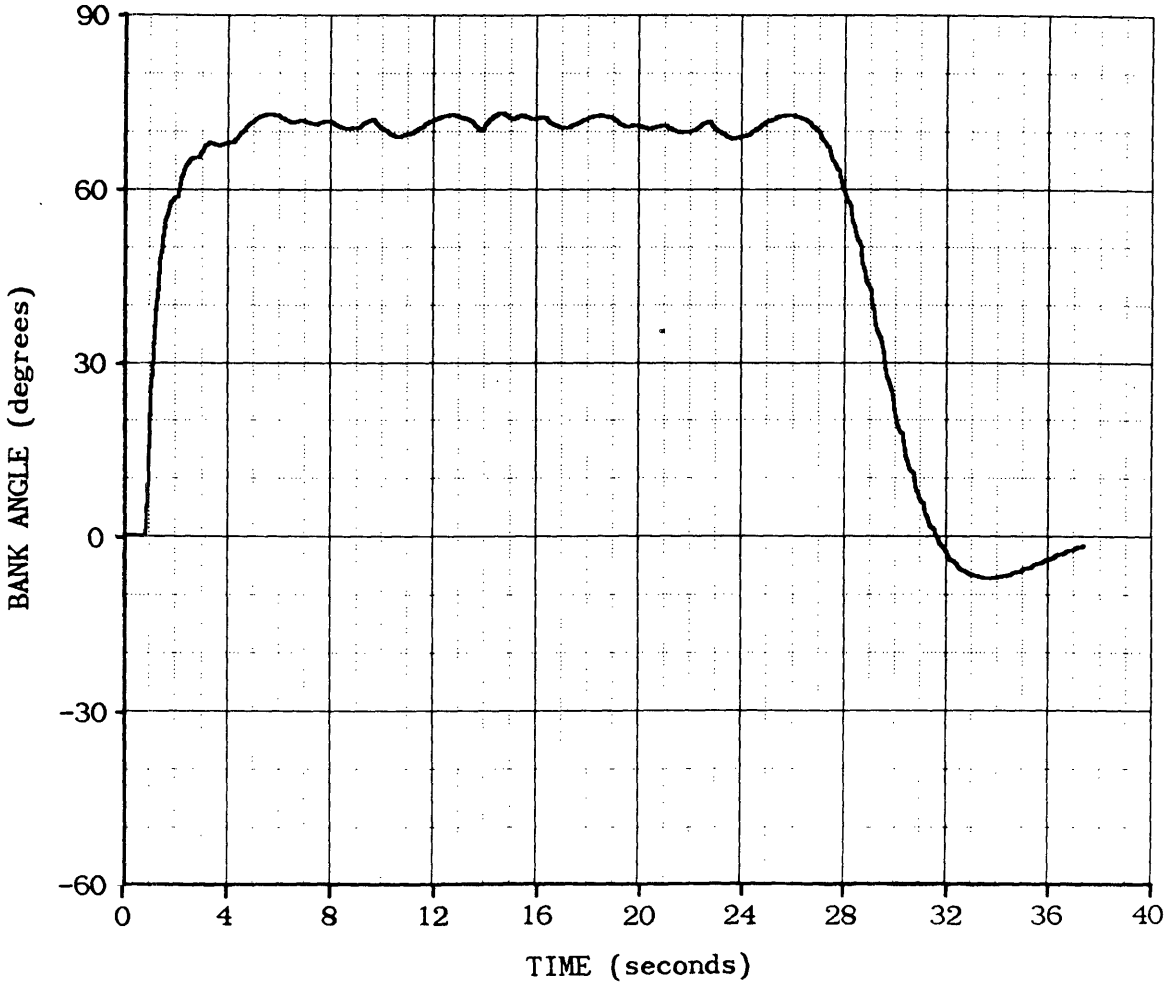

Figure 4.10 Aircraft Bank Angle vs Time for the Half-Circle Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index

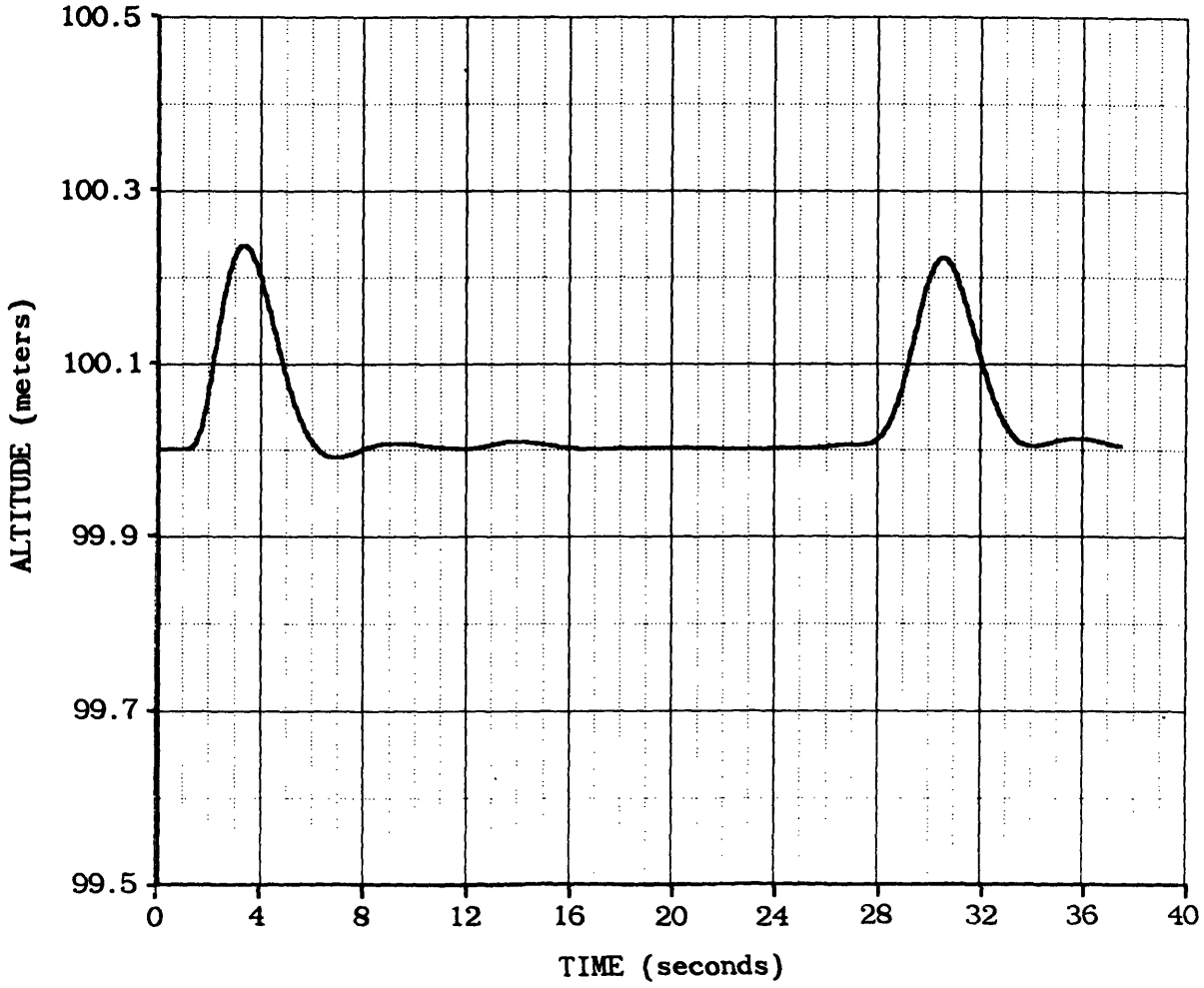

Figure 4.11 Aircraft Altitude vs Time for the Half-Circle Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index

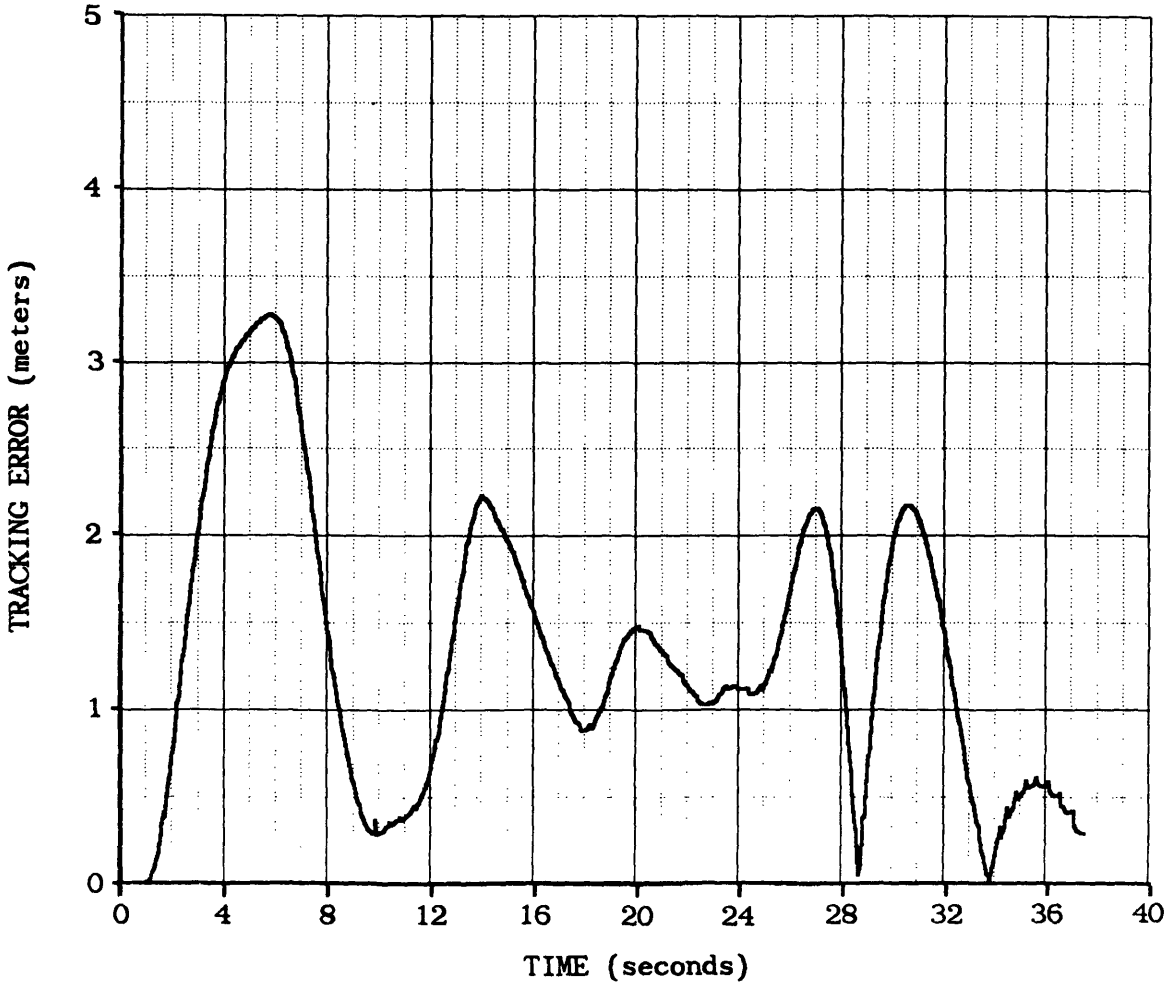

Figure 4.12 Aircraft Tracking Error vs Time for the Half-Circle Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index

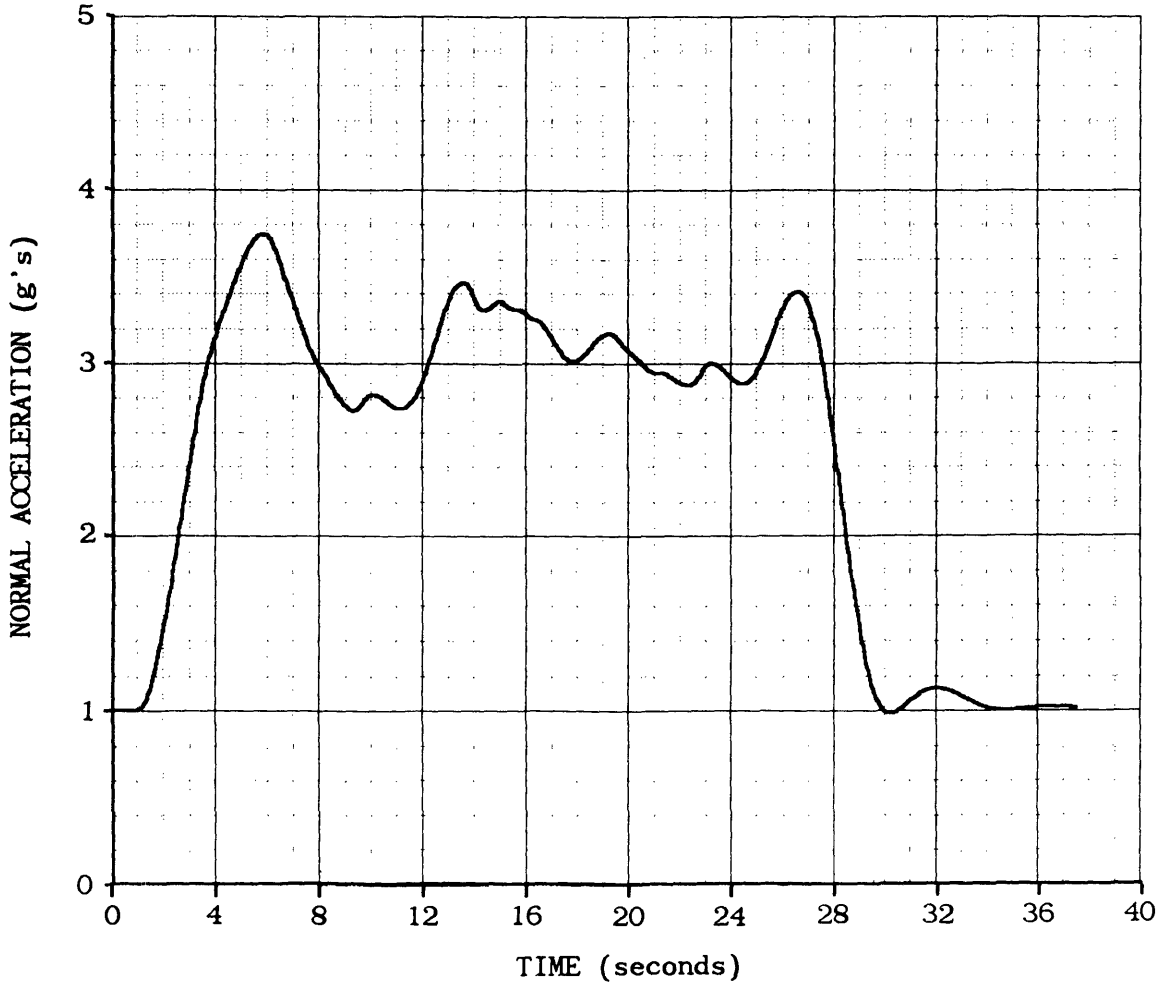

Figure 4.13 Aircraft Normal Acceleration vs Time for the Half-Circle Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index

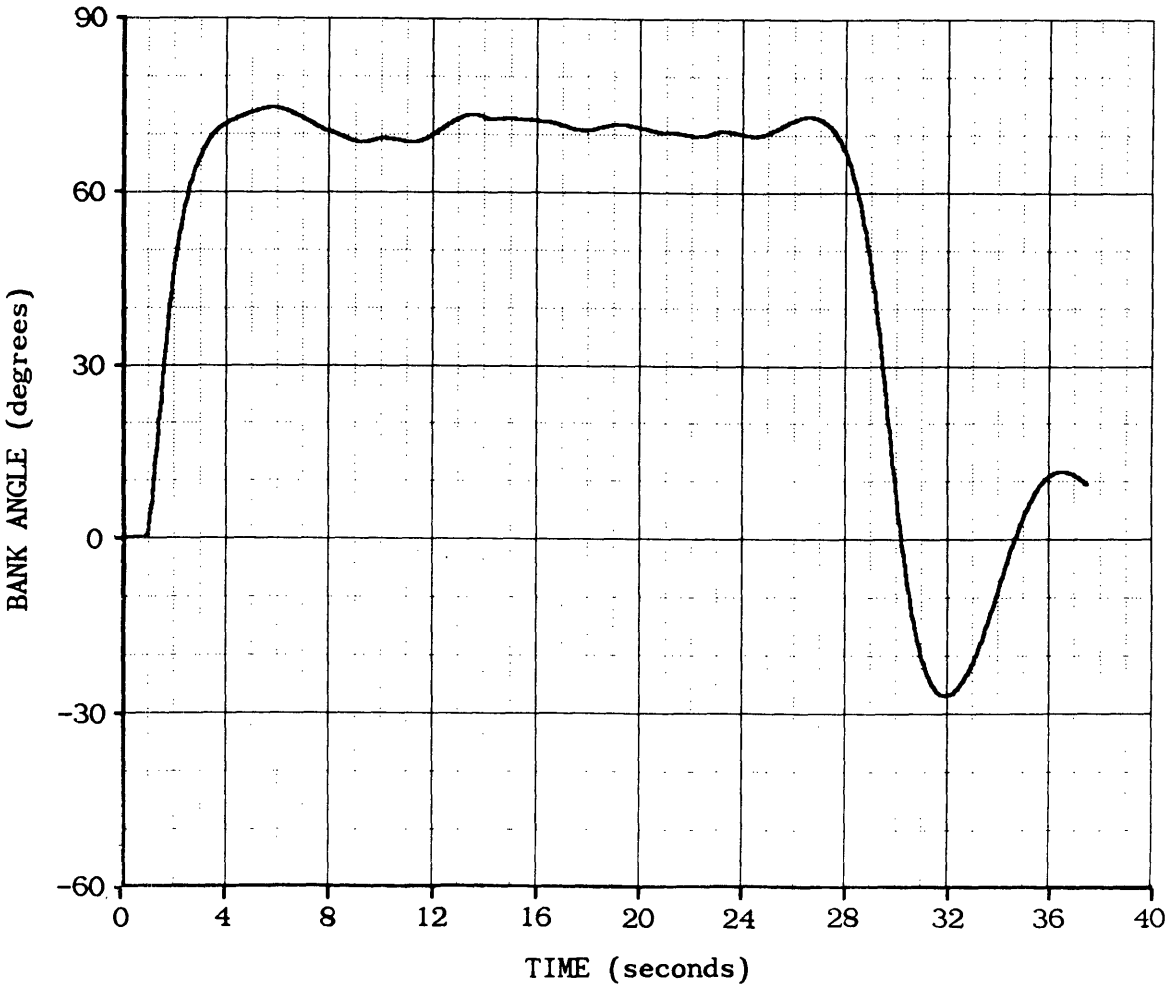

Figure 4.14 Aircraft Bank Angle vs Time for the Half-Circle Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index

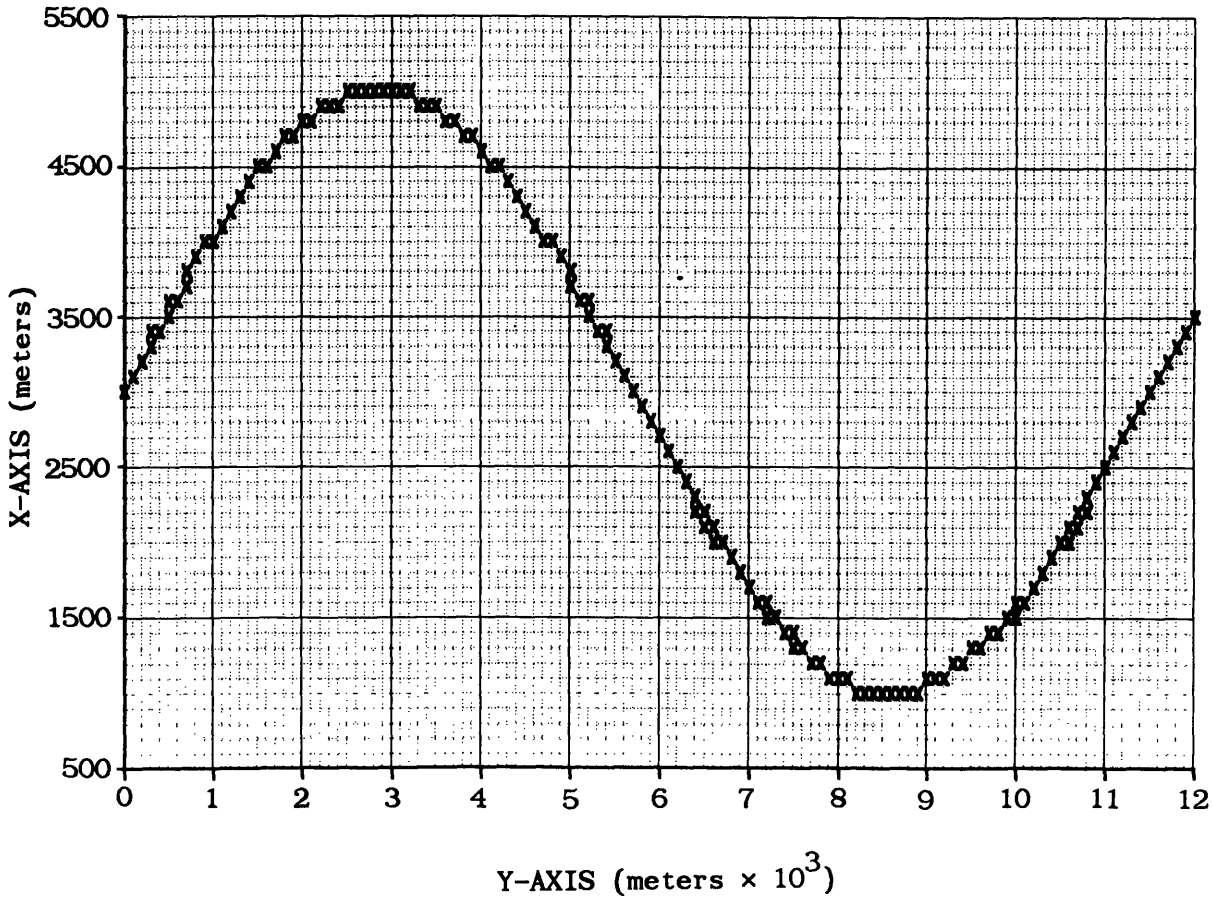

Figure 4.15 X-Y Reference Frame Coordinates for the Sine Curve Waypoint Set

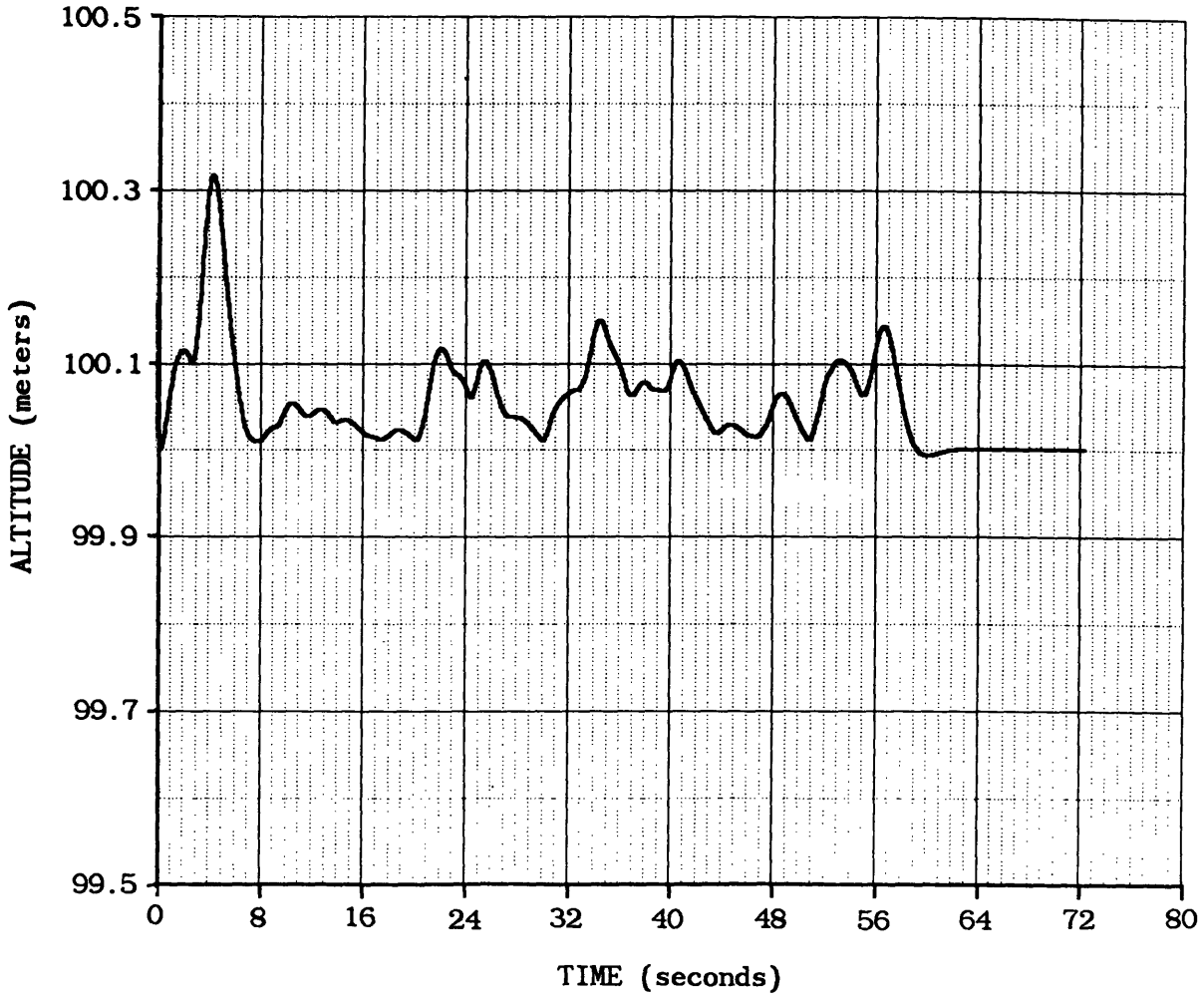

Figure 4.16 Aircraft Altitude vs Time for the Sine Curve Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index

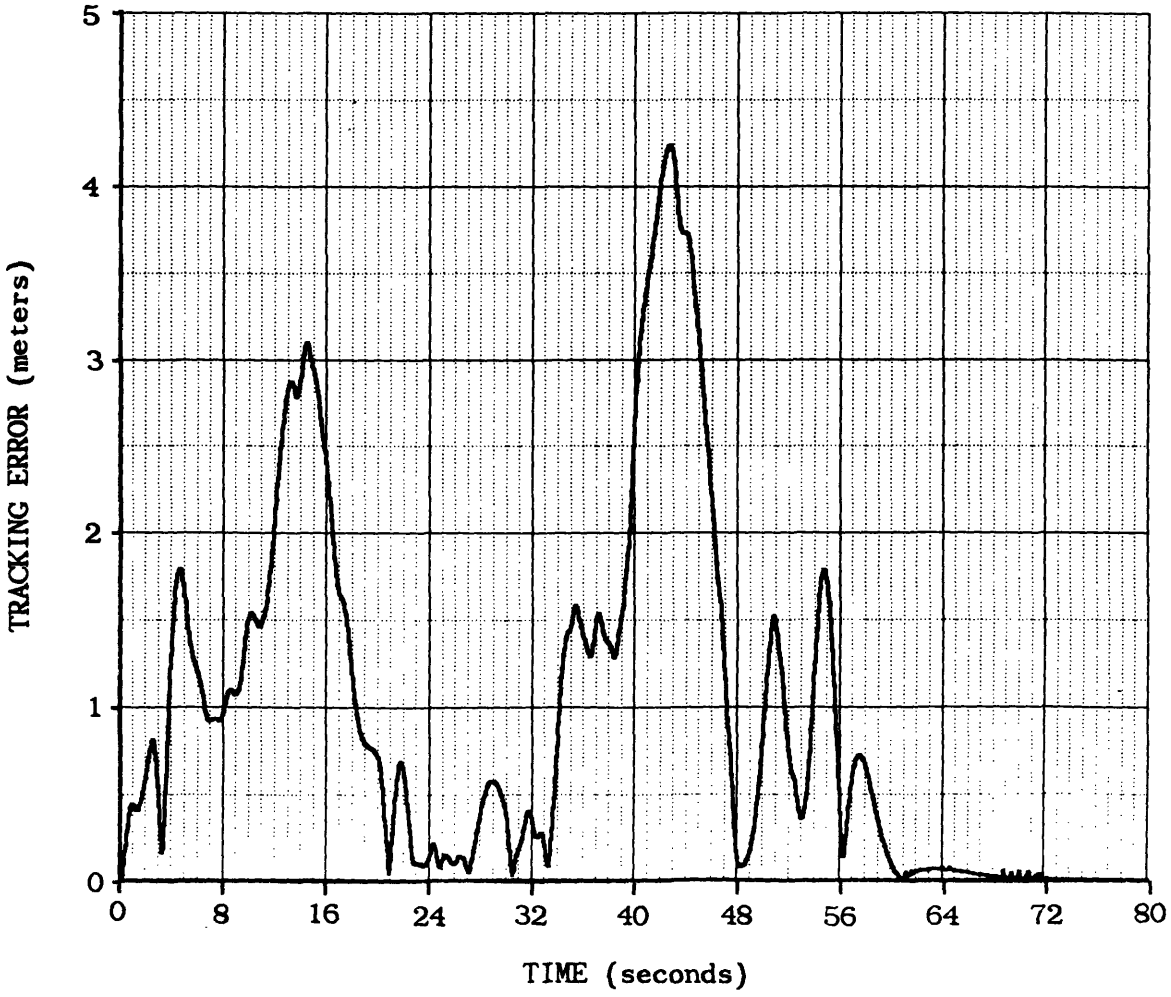

Figure 4.17 Aircraft Tracking Error vs Time for the Sine Curve Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index

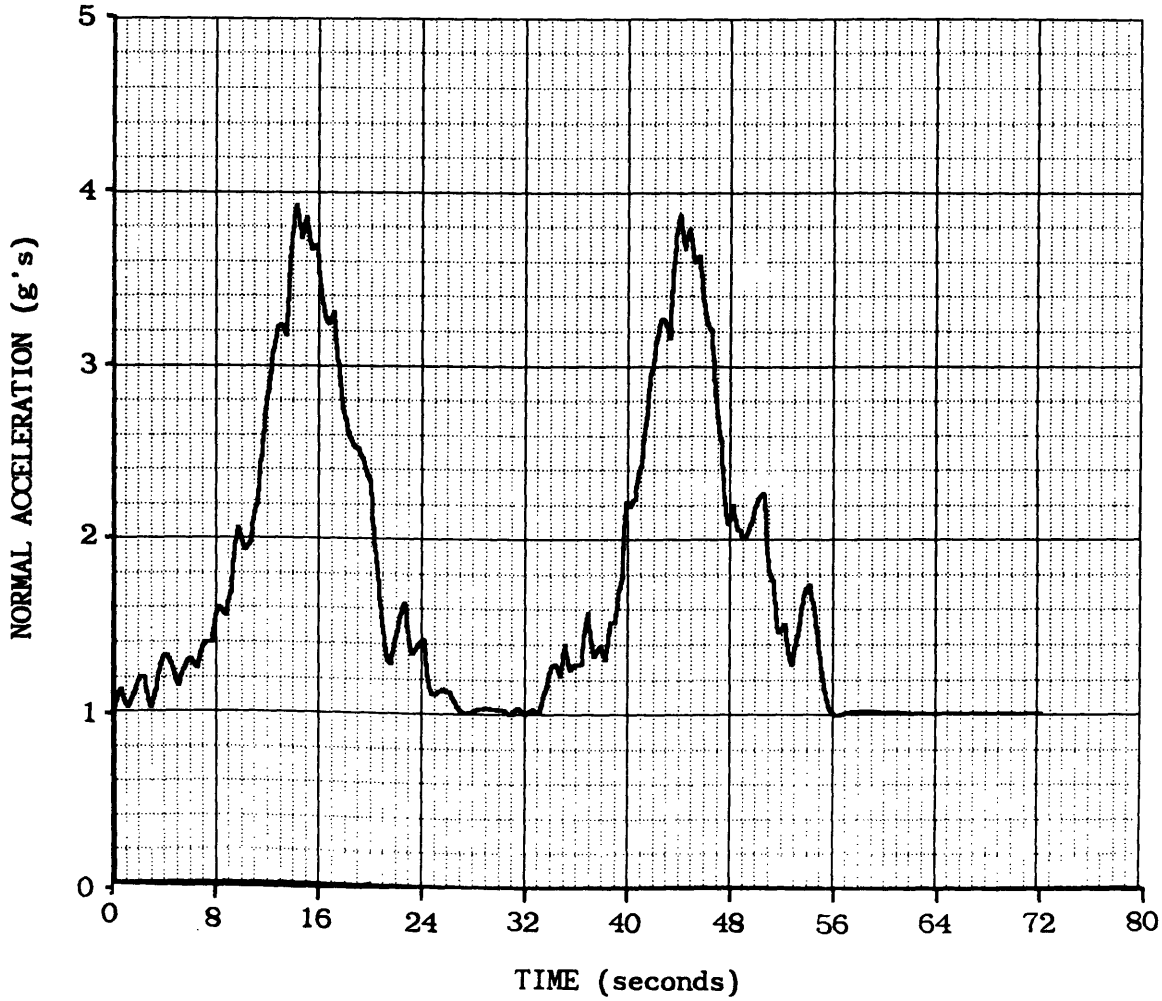

Figure 4.18 Aircraft Normal Acceleration vs Time for the Sine Curve Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index

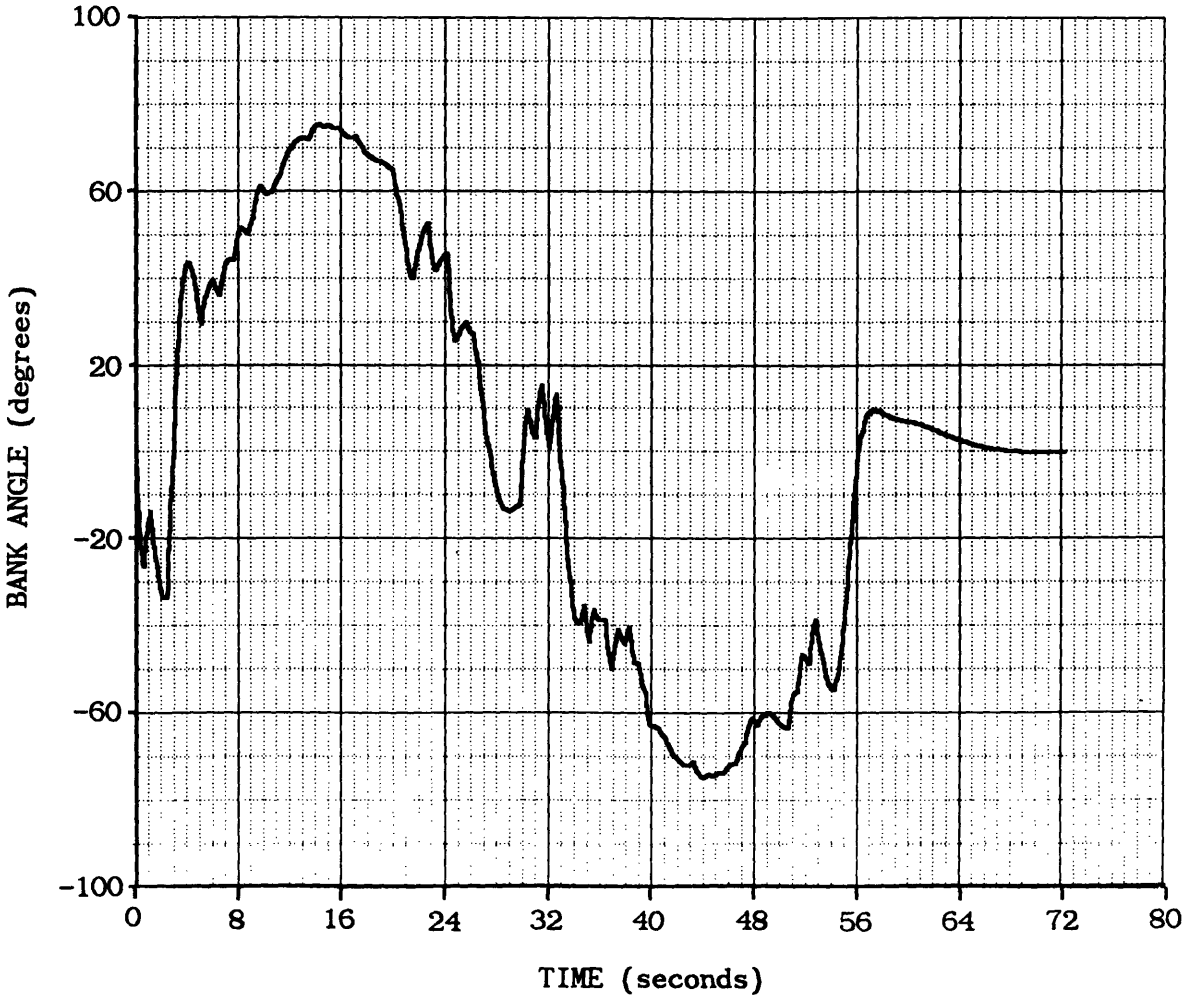

Figure 4.19 Aircraft Bank Angle vs Time for the Sine Curve Waypoint Set Using Constrained Second-Order Curve Fits in Subsets of Nine Matched in Position and Slope at the Second Subset Index

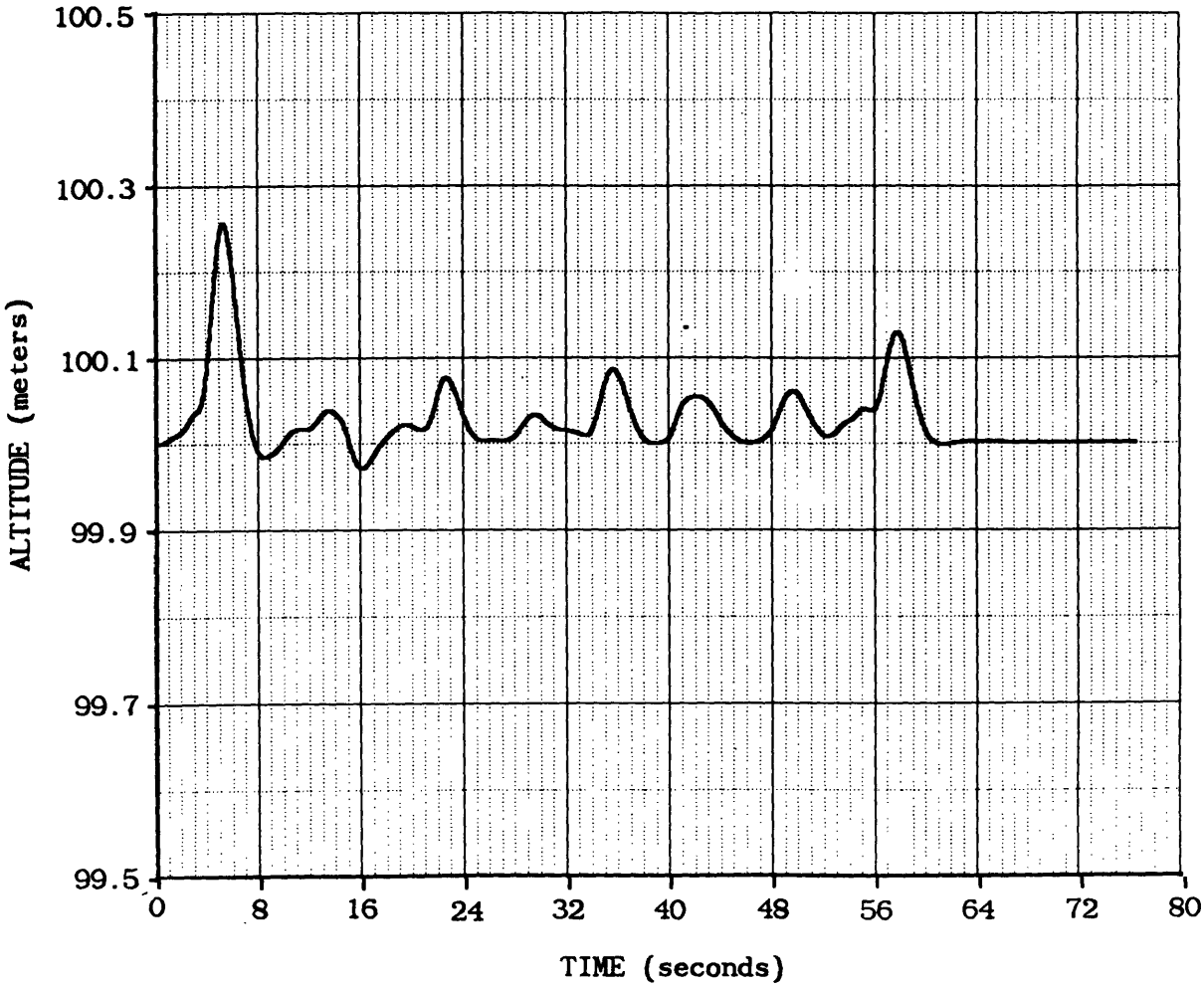

Figure 4.20 Aircraft Altitude vs Time for the Sine Curve Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index

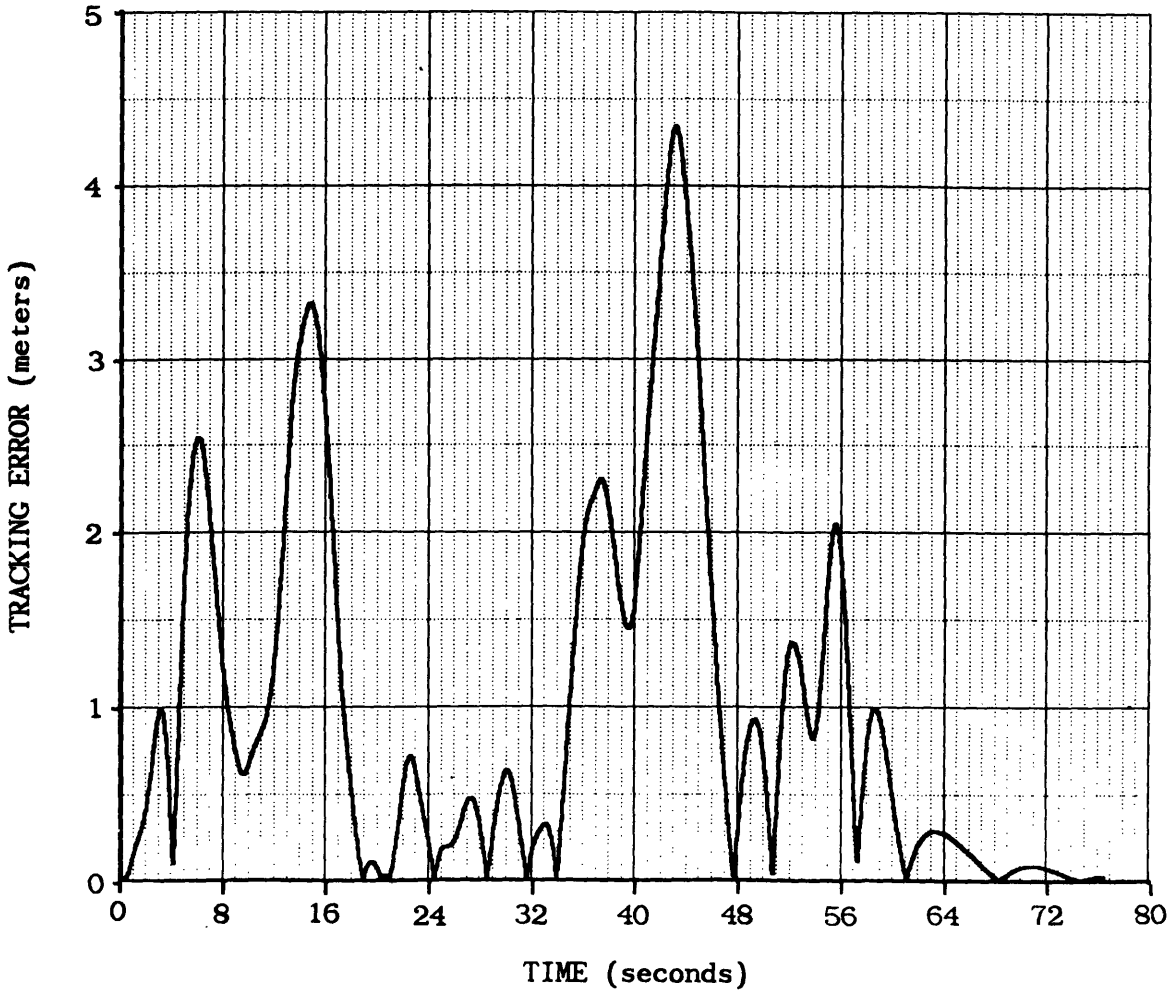

Figure 4.21 Aircraft Tracking Error vs Time for the Sine Curve Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index

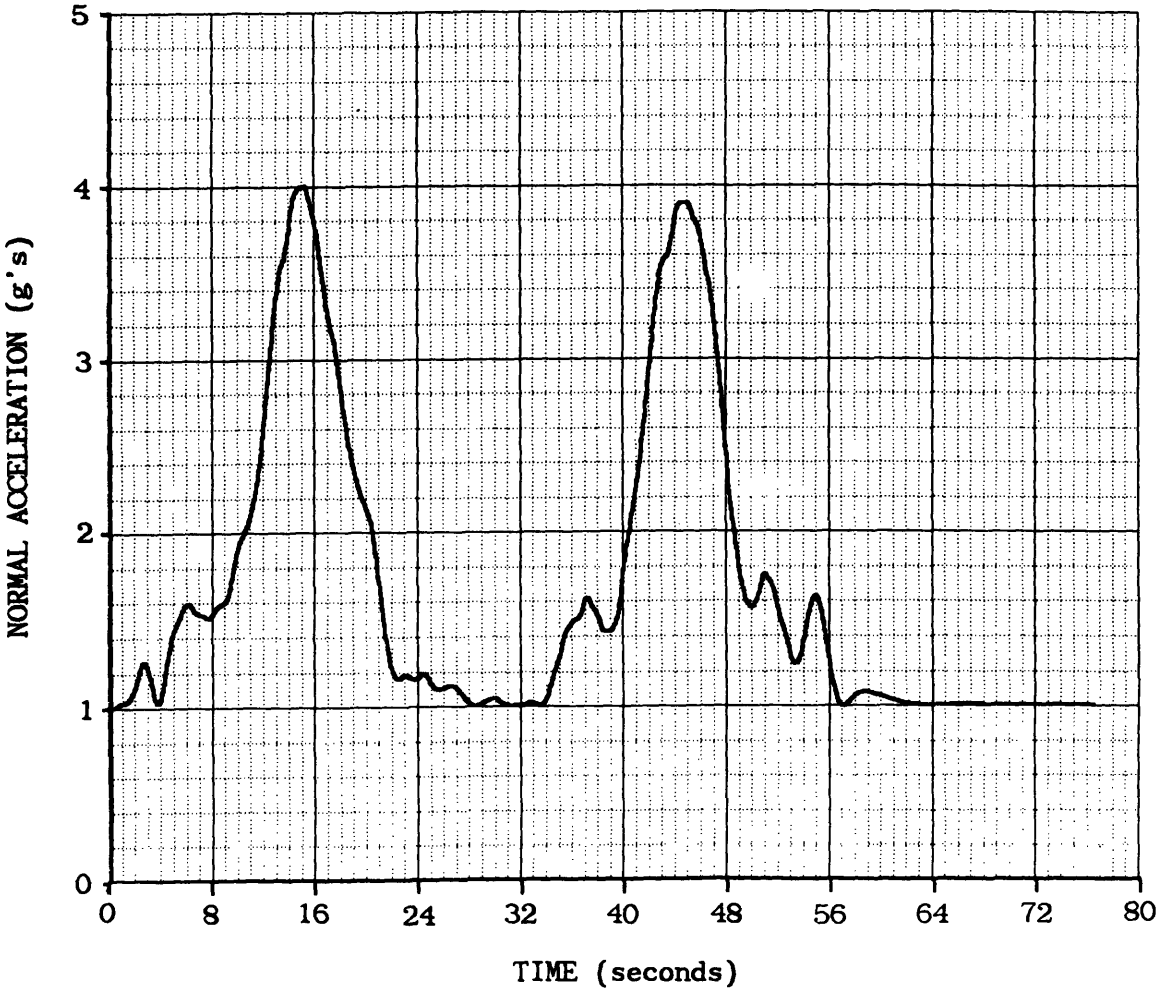

Figure 4.22 Aircraft Normal Acceleration vs Time for the Sine Curve Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index

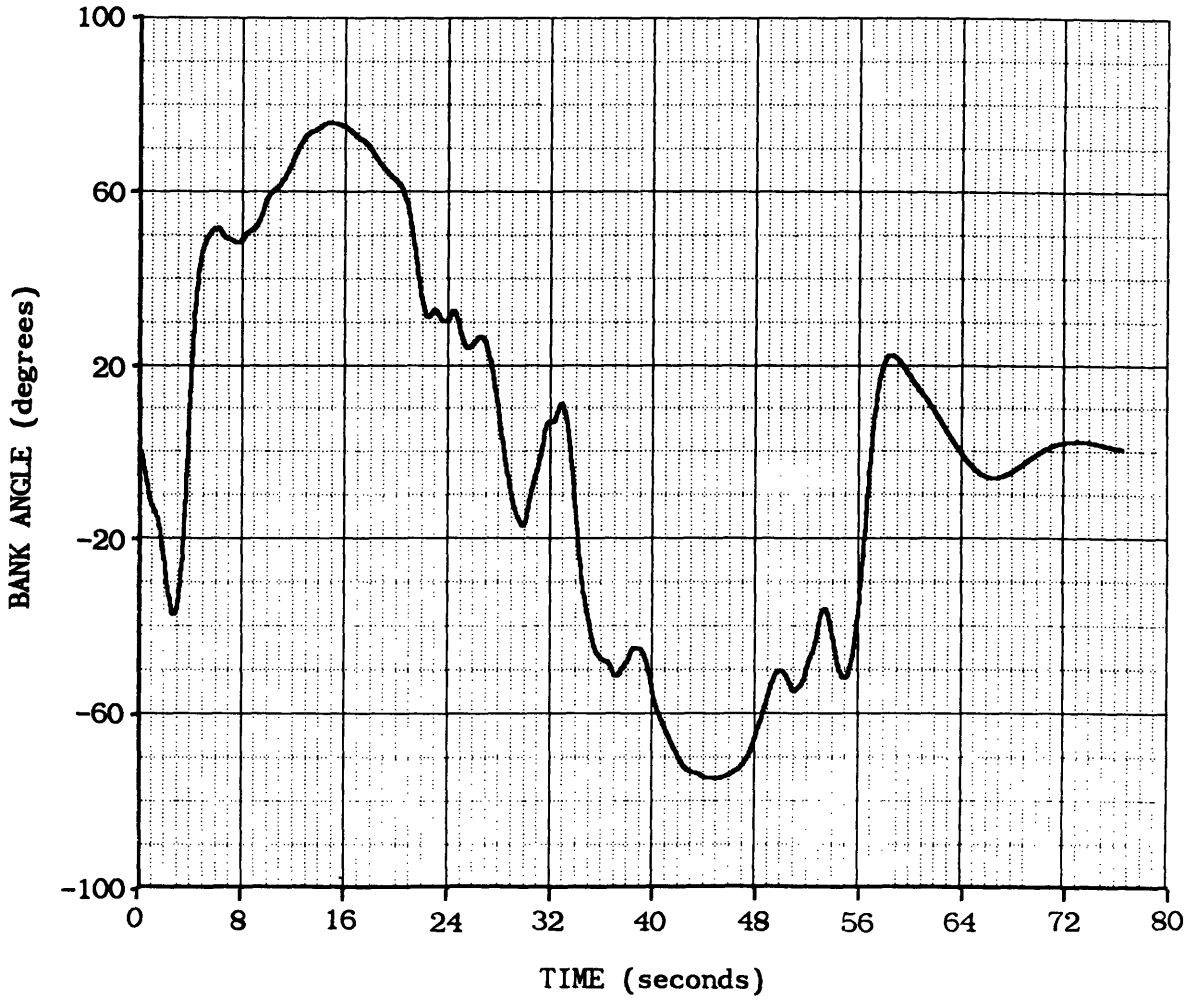

Figure 4.23 Aircraft Bank Angle vs Time for the Sine Curve Waypoint Set Using Constrained Third-Order Curve Fits in Subsets of Nine Matched in Position, Slope and Curvature at the Second Subset Index

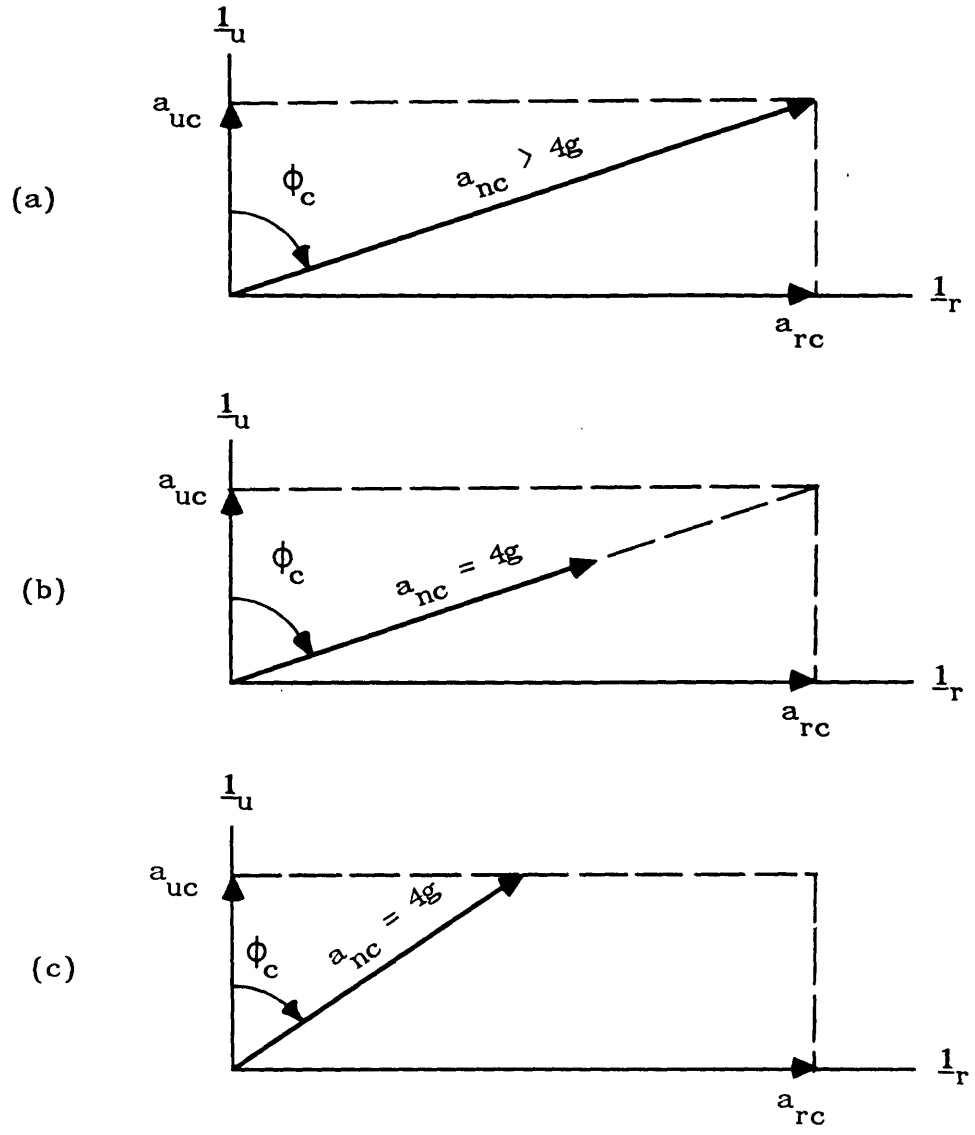

Figure 4.24 Aircraft Control Saturation (a) and Two Options for Command Limiting (b and c)

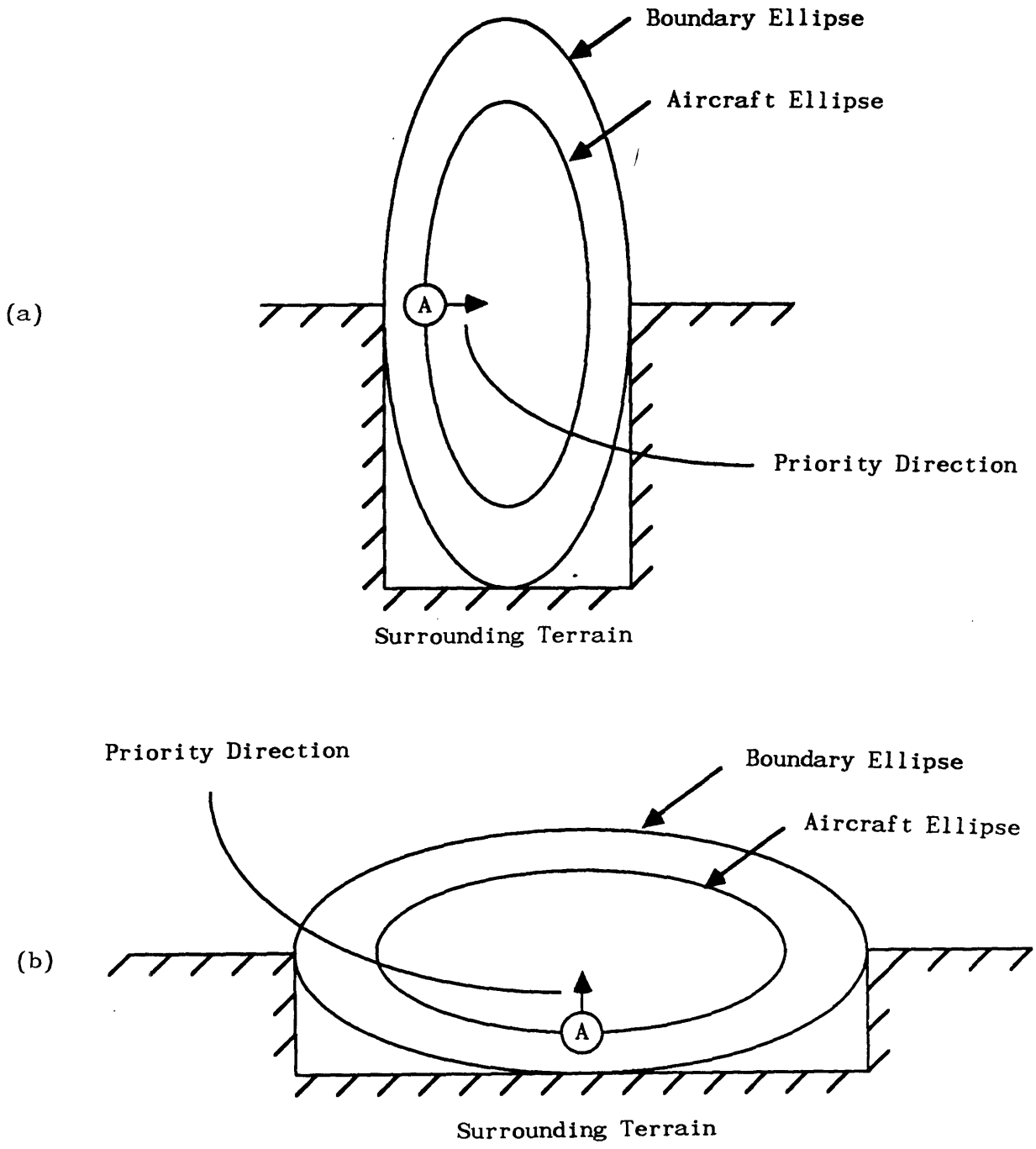

Figure 4.25 Illustration of Right (a) and Up (b) Priority Directions

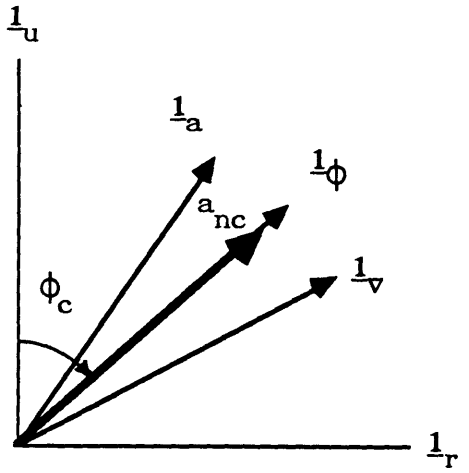

Figure 4.26 Generation of Normal Acceleration and Bank Angle Commands With Command Limiting

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

The objective of this thesis was to investigate a command interface between a trajectory planning system and a vehicle control system. Specifically, a command interface was devised to link an existing trajectory planning system to an aircraft FCS.

To begin the command interface development, models for the trajectory planning system and the aircraft FCS were developed. The trajectory planner is modeled as a static set of waypoints coordinatized in a common reference frame. Along with the waypoints, the planner provides terrain information in the form of ellipsoids that are centered at each waypoint. These ellipsoids describe the boundary terrain around each waypoint and are used by the command interface for control saturation logic. The aircraft FCS model contains two channels: normal acceleration and bank angle. The dynamics for each channel are given by a first-order lag which represents the aircraft time response to the FCS commands.

With the trajectory planner and FCS models, the nature and structure of the command interface were investigated and the required functions of the command interface were defined. The required functions of the interface are:

(1) Fit smooth curves through the waypoints to create a continuous desired trajectory

(2) Determine the position on the desired trajectory that is closest to the current aircraft position

(3) Using the closest trajectory position, compute the tracking error, the rate of change of tracking error, and the aircraft nominal acceleration

(4) Compute normal acceleration and bank angle commands for the aircraft FCS.

In its final form, the interface curve fitting uses sets of curves obtained from a third-order least-squares fitting procedure. These curves are formed by dividing the waypoint set into subsets of nine waypoints. These subsets are overlapped by seven waypoints, or "matched" at the second index in each subset, and are constrained to be continuous in position, slope, and curvature at the matching point.

The command generation algorithm developed to create the normal acceleration and bank angle commands uses a combination of tracking error, rate of change of tracking error, and aircraft nominal acceleration. The aircraft nominal acceleration is determined by calculating the acceleration which the aircraft would have if there were no tracking error. Additional logic was added to the command generation algorithm to account for control saturation and also to choose the proper sign (positive or negative) for the normal acceleration commands.

Through simulation the command interface performance was tested and found to be satisfactory for the scope of this work. The results are useful for determining the feasibility of computer-based vehicle controllers and are an important step toward eventual implementation of such systems.

It is important to note that many assumptions were made in the progress of this work. The major area that limits the application of these results is the model used for the trajectory planning system. As mentioned in Chapter 2, the trajectory planner was only simulated in a static configuration; actual implementation will obviously have the planner updating the vehicle's desired trajectory in real time. The use of discrete waypoints to represent trajectories creates the need for curve fitting algorithms in the command interface and thereby increases the required functionality of the interface.

In spite of the assumptions, the final form of the aircraft command interface presented here illustrates the feasibility of computer-based vehicle controllers. The navigational and aircraft state data required by the command interface are compatible with modern technology. Additionally, the computational burden implied by the command interface is not excessive when compared to the capability of current digital flight computers. For the level of complexity assumed for the overall system, the results of Chapter 4 show that the command interface performance is satisfactory.

5.2 Recommendations

The next step for this work would be to increase the complexity of the overall system by developing more rigorous and specific models for the trajectory planner and the aircraft FCS.

The method of representing trajectories by waypoints is not felt to be satisfactory since the first function the interface must perform is that of changing waypoints into smooth desired trajectories. The required interface functions can immediately be reduced if the trajectory planner is

forced to provide trajectories that can be directly used by the interface. Alternative methods of representing trajectories need to be developed and tested.

The complexity and capability of the aircraft FCS model needs to be increased in order for the aircraft maneuverability to be more realistic. In general, changing the FCS requires changing the command generation algorithm. Therefore, the FCS model and the command generation algorithm should be simultaneously developed to include a wider range of control commands and vehicle motions. This might include such aspects as aircraft thrust control or a command algorithm that accounts for uncoordinated flight conditions and wind.

The general structure of the command generation algorithm could also be changed to use other forms of feedback control. The tracking problem can be expanded to an optimal control structure by including time and energy constraints. This would allow the control generation to be affected by the urgency of given aircraft situations and could increase the overall effectiveness of a computer-based aircraft controller.

Finally, further static and dynamic (in the sense of on-line trajectory planning) simulation needs to be done in order to fully understand the interactions between the trajectory planner and the vehicle control system. Man-rated vehicle constraints and their effects upon the command interface design should also be investigated and addressed.

References

[1] Daly, K.; Motyka, P.; Nurse, R.; Palmer, P.; Schmidt, G. "Development of Capability for Multifunction Integrated Reference Assembly Evaluation," Charles Stark Draper Laboratory, Inc., July 1977.

[2] James, M.L.; Smith, G.M.; Wolford, J.C. Applied Numerical Methods for Digital Computation, Third Edition. Harper and Row, 1985.

[3] Thomas, G.B. Jr. Calculus and Analytic Geometry, Third Edition. Addison-Wesley, 1966.

[4] Etkin, B. Dynamics of Flight: Stability and Control. John Wiley & Sons, 1959.

Appendix A

A.1 Half-Circle Waypoint Set

Description:

The waypoints describe a right-turning half-circle in the X-Y reference plane. All of the waypoints are at an altitude of 100 meters (Z = -100 m). Table A.1 gives the X-Y coordinates in the reference frame of the waypoints in the set.

Aircraft Initial Conditions:

X position = 100 m	X velocity = 240 m/s
Y position = 500 m	Y velocity = 0 m/s
Z position = -100 m	Z velocity = 0 m/s

Normal Acceleration = 9.81 m/s^2
Bank Angle = 0 deg

A.2 Sine Curve Waypoint Set

Description:

The waypoints represent one cycle of a sine curve in the X-Y reference plane. All of the waypoints are at an altitude of 100 meters. Table A.2 gives the X-Y coordinates in the reference plane of the waypoints in the set.

Aircraft Initial Conditions:

X position = 3000 m	X velocity = 170 m/s
Y position = 0 m	Y velocity = 170 m/s
Z position = -100 m	Z velocity = 0 m/s

Normal Acceleration = 9.81 m/s^2
Bank Angle = 0 deg

Table A.1 Reference Frame X-Y Coordinates for the Half-Circle Waypoint Set (in meters)

Waypoint index	X	Y	Waypoint index	X	Y
1	100	500	46	2200	3700
2	200	500	47	2100	3800
3	300	500	48	2000	3900
4	400	500	49	1900	4000
5	500	500	50	1800	4100
6	600	500	51	1700	4200
7	700	500	52	1600	4200
8	800	500	53	1500	4300
9	900	500	54	1400	4300
10	1000	500	55	1300	4400
11	1100	600	56	1200	4400
12	1200	600	57	1100	4400
13	1300	600	58	1000	4500
14	1400	700	59	900	4500
15	1500	700	60	800	4500
16	1600	800	61	700	4500
17	1700	800	62	600	4500
18	1800	900	63	500	4500
19	1900	1000	64	400	4500
20	2000	1100	65	300	4500
21	2100	1200	66	200	4500
22	2200	1300	67	100	4500
23	2300	1400	68	000	4500
24	2300	1500	69	-100	4500
25	2400	1600	70	-200	4500
26	2400	1700	71	-300	4500
27	2500	1800	72	-400	4500
28	2500	1900	73	-500	4500
29	2500	2000	74	-600	4500
30	2600	2100	75	-700	4500
31	2600	2200	76	-800	4500
32	2600	2300	77	-900	4500
33	2600	2400	78	-1000	4500
34	2600	2500	79	-1100	4500
35	2600	2600	80	-1200	4500
36	2600	2700	81	-1300	4500
37	2600	2800	82	-1400	4500
38	2600	2900	83	-1500	4500
39	2500	3000	84	-1600	4500
40	2500	3100	85	-1700	4500
41	2500	3200	86	-1800	4500
42	2400	3300	87	-1900	4500
43	2400	3400	88	-2000	4500
44	2300	3500	89	-2100	4500
45	2300	3600	90	-2200	4500

Table A.2 Reference Frame X-Y Coordinates for the Sine Curve Waypoint Set (meters)

Waypoint index	X	Y	Waypoint index	X	Y
1	3000	000	46	4500	4200
2	3100	100	47	4400	4300
3	3200	200	48	4300	4400
4	3300	300	49	4200	4500
5	3400	300	50	4100	4600
6	3400	400	51	4000	4700
7	3500	500	52	4000	4800
8	3600	500	53	3900	4900
9	3600	600	54	3800	5000
10	3700	700	55	3700	5000
11	3800	700	56	3600	5100
12	3900	800	57	3600	5200
13	4000	900	58	3500	5200
14	4000	1000	59	3400	5300
15	4100	1100	60	3400	5400
16	4200	1200	61	3300	5400
17	4300	1300	62	3200	5500
18	4400	1400	63	3100	5600
19	4500	1500	64	3000	5700
20	4500	1600	65	2900	5800
21	4600	1700	66	2800	5900
22	4700	1800	67	2700	6000
23	4700	1900	68	2600	6100
24	4800	2000	69	2500	6200
25	4800	2100	70	2400	6300
26	4900	2200	71	2300	6400
27	4900	2300	72	2200	6400
28	4900	2400	73	2200	6500
29	5000	2500	74	2100	6500
30	5000	2600	75	2100	6600
31	5000	2700	76	2000	6600
32	5000	2800	77	2000	6700
33	5000	2900	78	1900	6800
34	5000	3000	79	1800	6900
35	5000	3100	80	1700	7000
36	5000	3200	81	1600	7100
37	4900	3300	82	1600	7200
38	4900	3400	83	1500	7200
39	4900	3500	84	1500	7300
40	4800	3600	85	1400	7400
41	4800	3700	86	1400	7500
42	4700	3800	87	1300	7500
43	4700	3900	88	1300	7600
44	4600	4000	89	1200	7700
45	4500	4100	90	1200	7800

Table A.2 (Continued)

Waypoint index	X	Y	Waypoint index	X	Y
91	1100	7900	114	1600	10100
92	1100	8000	115	1700	10200
93	1100	8100	116	1800	10300
94	1000	8200	117	1900	10400
95	1000	8300	118	2000	10500
96	1000	8400	119	2000	10600
97	1000	8500	120	2100	10600
98	1000	8600	121	2100	10700
99	1000	8700	122	2200	10700
100	1000	8800	123	2200	10800
101	1000	8900	124	2300	10800
102	1100	9000	125	2400	10900
103	1100	9100	126	2500	11000
104	1100	9200	127	2600	11100
105	1200	9300	128	2700	11200
106	1200	9400	129	2800	11300
107	1300	9500	130	2900	11400
108	1300	9600	131	2000	11500
109	1400	9700	132	3100	11600
110	1400	9800	133	3200	11700
111	1500	9900	134	3300	11800
112	1500	10000	135	3400	11900
113	1600	10000	136	3500	12000